

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta strojního inženýrství

DIPLOMOVÁ PRÁCE

Brno, 2016

Bc. Jiří Černota



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

LETECKÝ ÚSTAV

INSTITUTE OF AEROSPACE ENGINEERING

**VYUŽITÍ PROSTŘEDÍ MATLAB SIMULINK PŘI VÝUCE
MECHANIKY LETU II**

MATLAB SIMULINK FOR FLIGHT MECHANICS II MODULE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Jiří Černota

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jiří Matějů

BRNO 2016

Zadání diplomové práce

Ústav: Letecký ústav
Student: **Bc. Jiří Černota**
Studijní program: Strojní inženýrství
Studijní obor: Stavba letadel
Vedoucí práce: **Ing. Jiří Matějů**
Akademický rok: 2015/16

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Využití prostředí Matlab Simulink při výuce Mechaniky letu II

Stručná charakteristika problematiky úkolu:

Znalost práce v prostředí Matlab a Matlab Simulink se stává základní dovedností inženýra. Mechanika letu II je vhodný předmět pro procvičení práce v tomto prostředí. Student se seznámí s prostředím Matlab, Matlab Simulink a vytvoří vhodný příklad do cvičení Mechaniky letu II. Práce může být později využita pro praktické výpočty letových vlastností reálného letounu.

Cíle diplomové práce:

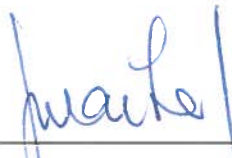

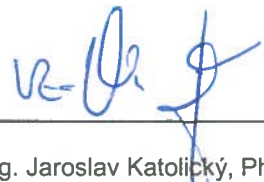
Příklad pro cvičení Mechaniky letu II.

Seznam literatury:

Mathworks, <http://www.mathworks.com>, 3. října 2015.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2015/16.

V Brně, dne 2. 12. 2015

		
_____ doc. Ing. Jaroslav Juračka, Ph.D. ředitel ústavu		_____ doc. Ing. Jaroslav Katolický, Ph.D. děkan

ABSTRAKT

Tato diplomová práce se zabývá využitím softwaru Matlab Simulink při výuce předmětu Mechanika letu II. Pro účely této práce bylo vytvořeno uživatelské prostředí umožňující výpočet aerodynamických derivací a matic stavového prostoru, které zároveň slouží ke spouštění simulací dynamiky podélného pohybu letounu v prostředí Simulink a vizualizaci jejich výsledků. V poslední části práce byly navrženy příklady do cvičení ze zmíněného předmětu, zaměřené na osvojení si základů blokového programování a vytvoření představy o dynamice a řízení pohybu letounu.

KLÍČOVÁ SLOVA

Matlab, Simulink, Mechanika letu, FDC, Stabilitní derivace

ABSTRACT

The thesis is about the use of Matlab Simulink software for subject Flight Mechanics II. User interface, which allow computing of aerodynamic derivations and state space matrices, was created. It also provides system for launching simulations of longitudinal aircraft movement in Simulink environment and results visualization. In the last part were suggested examples for practice lessons of the aforementioned subject. These lessons focus on mastering basic block programming and to understand concept of aircraft dynamics and controlling aircraft movement.

KEYWORDS

Matlab, Simulink, Flight dynamic, FDC, Stability derivative

ČERNOTA, Jiří *Využití prostředí Matlab Simulink při výuce Mechaniky letu II*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Letecký ústav, 2016. 67 s. Vedoucí práce byl Ing. Jiří Matějů

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Využití prostředí Matlab Simulink při výuce Mechaniky letu II“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu práce Ing. Jiřímu Matěju za odborné rady a pomoc při zpracování této práce. Dále děkuji Bc. Adriánu Martincovi za technickou pomoc při programování v programu Matlab.

Brno

.....

(podpis autora)

OBSAH

Úvod	8
1 Teoretický základ	9
1.1 Matematický popis letounu	10
1.1.1 Souřadné soustavy	10
1.1.2 Pohybové rovnice	12
1.2 Letoun jako dynamický systém	14
2 Matlab	16
2.1 Programovací prostředí Matlab a Simulink	16
2.1.1 Uživatelské prostředí Matlabu	16
2.1.2 Matlab simulink	17
2.1.3 Matlab guide	19
3 FDC	21
3.1 Teoretické pozadí nelineárního modelu	21
3.1.1 Vnější síly a momenty	24
3.2 Implementace nelineárního modelu FDC	25
4 VDVL	28
4.1 Základní uživatelské rozhraní	28
4.2 Zadávání a výpočet	30
4.2.1 Ukládání a načítání dat z databáze	32
4.3 Propojení s Matlab Simulink	33
4.3.1 Předdefinované modely na základě stavového prostoru	33
4.3.2 Předdefinovaný model na základě FDC	37
4.4 Vizualizace výsledků	38
5 Implementace programu VDVL	41
5.1 Inicializace programu	41
5.2 Uživatelské prostředí	42
5.3 Výpočty parametrů letounu	43
6 Využití VDVL ve výuce předmětu Mechanika letu II	46
6.1 Příklady do výuky	46
6.1.1 Model stavového prostoru	46
6.1.2 Regulátor úhlu θ	48

7 Závěr	51
Literatura	52
Seznam symbolů, veličin a zkratk	53
Seznam obrázků	55
Seznam tabulek	57
Seznam příloh	58
A Seznam zkratk v programu VDVL	59
B Tutorial k programu VDVL	62

ÚVOD

Program Matlab Simulink nachází čím dál větší uplatnění ve většině technických oborech, kde je zapotřebí vytvářet programy pro simulace odezev dynamických systémů. Díky jednoduchému interaktivnímu prostředí a intuitivnímu způsobu blokového programování se Matlab Simulink stává vhodnou pomůckou také při výukách na technicky zaměřených školách. Zvláště v předmětu Mechanika letu II má potenciál pomáhat studentům při osvojování si vlivu některých parametrů letounu na jeho dynamické letové vlastnosti.

V první části práce je nastíněna problematika matematického popisu letových vlastností letounu a vytváření modelů pro dynamické simulace. Dále je stručně popsáno programovací prostředí Matlab a Matlab Simulink. Je zde také zmíněn model pro řešení dynamických vlastností a řízení letounu FDC, který je dále použit v další části práce.

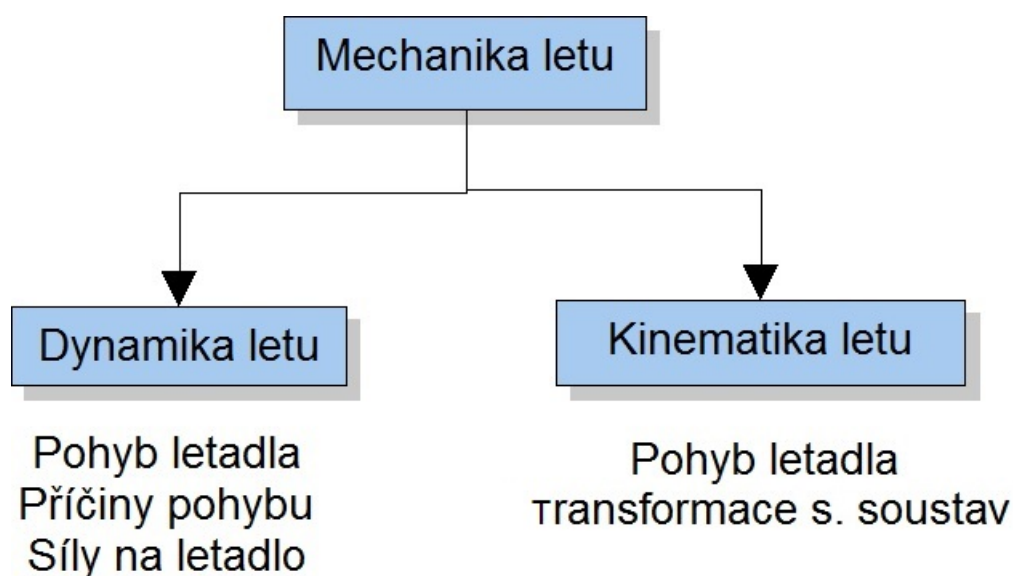
Ve druhé části je popsán program VDVL, který byl vytvořen v rámci praktické části diplomové práce. Je zde popsáno jeho uživatelské prostředí a implementace ve skriptovacím programovacím jazyku Matlab.

Poslední část se zabývá využitím programu VDVL ve výuce zmíněného předmětu a návrhem příkladů do cvičení, založených na práci s programem a Matlab Simulinkem.

1 TEORETICKÝ ZÁKLAD

"Mechanika letu je aplikovaná fyzikálně-technická disciplína zkoumající zákonitosti pohybu letadel v souvislosti s příčinami, které pohyb způsobují." [2].

Mechaniku letu atmosférického letounu lze rozdělit do dvou základních oddílů jak je naznačeno na obrázku 1.1. Kinematika letu se zabývá pohybem letounu v prostoru a čase a jeho popisem v různých souřadnicových soustavách bez ohledu na působící síly. Dynamika letu zkoumá úplný pohyb letadla v souvislosti s příčinami, které tento pohyb způsobují. K tomuto využívá poznatky ze základní mechaniky tuhých těles spolu s dalšími disciplínami jako jsou aerodynamika, teorie řízení a regulace [2].



Obr. 1.1: Rozdělení mechaniky letu atmosférického letounu

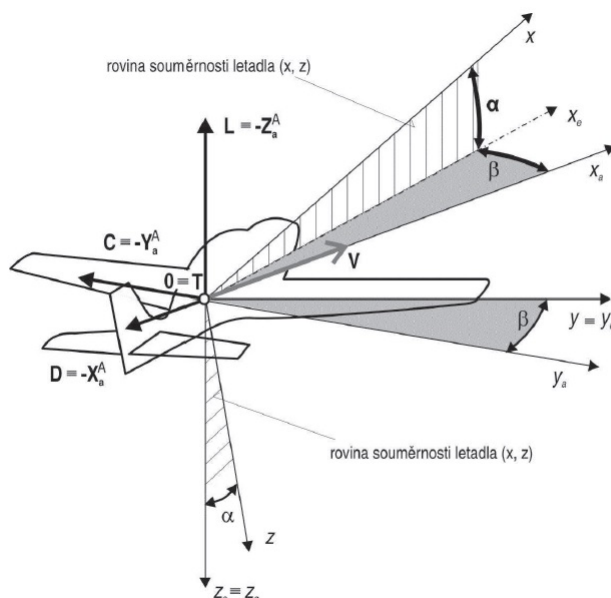
Zaměřením se na matematický popis pohybu letounu a silového působení na něj je způsob jak kvantifikovat jeho letové vlastnosti. To umožňuje porovnávat stabilitu a říditelnost letounů a zanášet objektivní požadavky na tyto vlastnosti do předpisů, což ve svém důsledku zvyšuje bezpečnost letectví. Výsledky rozborů letounu z hlediska mechaniky letu jsou velmi důležité jak při návrhu letadla z hlediska letových vlastností a stability, tak při provozu z hlediska optimálního využití letounu. Nelze opomenout ani váhu kterou má pro piloty od počátečních fází výcviku až po každodenní pilotní praxi [2].

1.1 Matematický popis letounu

Základním úkolem matematického modelu letounu je popsat chování letounu ve zkoumaném režimu letu. Tyto režimy mohou být ustálené, kdy hovoříme o stavu letounu který se bez změny podmínek za jakých se ustálil nemění, nebo neustálené, kdy se stavové hodnoty samovolně mění v čase. Neustálený let může dále být stabilní, kdy má letoun snahu přejít do ustáleného letu, nebo nestabilní, pokud se zvyšuje míra neustálenosti [2, 1].

1.1.1 Souřadné soustavy

Pohyb jakéhokoli tělesa je možné sledovat jen vůči jiným tělesům, s nimiž spojujeme vztažnou souřadnicovou soustavu. Obecný prosotorový pohyb pohyb letadel popisujeme v určitých, pro danou úlohu vhodně zvolených, souřadnicových soustavách. Mezi nejčastěji používané patří zemská souřadnicová soustava pevně spojená se zemí, letadlová souřadnicová soustava pevně spojená s letadlem a aerodynamická souřadnicová soustava 1.2. Za zmínku stojí ještě stabilitní souřadnicová soustava, která je zvláštním případem letadlové soustavy. Ta se používá pro zkoumání dynamické stability a její výhoda spočívá v tom, že po poruše z rovnovážného letu pracujeme pouze s odchylkami kinematických veličin od referenčního režimu [1, 2, 3].



Obr. 1.2: Aerodynamická souřadnicová soustava (index a) a letadlová souřadnicová soustava (bez indexu) [2]

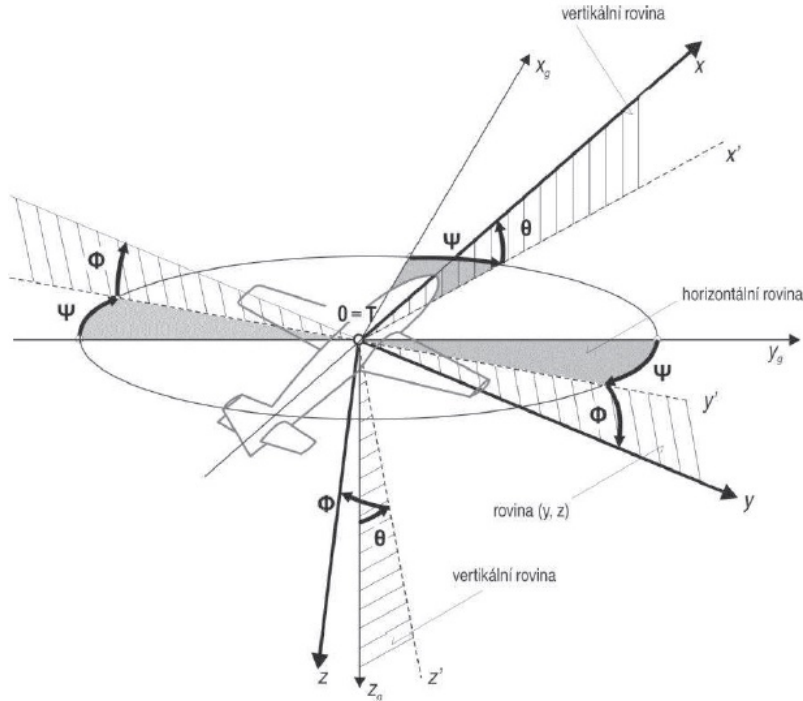
V úlohách mechaniky letu je často potřeba přecházet z jedné soustavy do druhé. K tomu slouží transformační matice 4.1 [2].

$$[T] = \begin{bmatrix} \cos\varphi_1 \cdot \cos\varphi_2 & \sin\varphi_1 \cdot \cos\varphi_2 & -\sin\varphi_2 \\ \begin{pmatrix} \cos\varphi_1 \cdot \sin\varphi_2 \cdot \sin\varphi_3 \\ -\sin\varphi_1 \cdot \cos\varphi_3 \\ \cos\varphi_1 \cdot \sin\varphi_2 \cdot \cos\varphi_3 \\ +\sin\varphi_1 \cdot \sin\varphi_3 \end{pmatrix} & \begin{pmatrix} \sin\varphi_1 \cdot \sin\varphi_2 \cdot \sin\varphi_3 \\ +\cos\varphi_1 \cdot \cos\varphi_3 \\ \sin\varphi_1 \cdot \sin\varphi_2 \cdot \cos\varphi_3 \\ +\cos\varphi_1 \cdot \sin\varphi_3 \end{pmatrix} & \begin{pmatrix} \cos\varphi_2 \cdot \sin\varphi_1 \\ \cos\varphi_2 \cdot \cos\varphi_3 \end{pmatrix} \end{bmatrix} \quad (1.1)$$

Hodnoty φ_1 , φ_2 a φ_3 v matici nahradíme polohovými úhly ve smluvně stanoveném pořadí a transformace složek vektoru z jedné soustav do druhé je pak vyjádřena vztahem 1.2.

$$\{S_2\} = [T] \{S_1\} \quad (1.2)$$

Eulerovy polohové úhly (Ψ , Θ , Φ) definující polohu letadla v prostoru vůči místní horizontální rovině (x_g , y_g) procházející těžištěm letadla jsou na obrázku 1.3. Podle souřadnicové soustavy, ve které se zrovna pohybujeme dále definujeme aerodynamické polohové úhly (χ_a , γ_a , μ_a) nebo dráhové polohové úhly [2].



Obr. 1.3: Eulerovy polohové úhly [2]

1.1.2 Pohybové rovnice

Zavedením některých zjednodušení jako je zanedbání pružnosti letecké konstrukce lze degradovat letoun na tuhý hmotný bod se šesti stupni volnosti a popisovat ho pomocí základních Newtonových pohybových rovnic 1.3 a 1.4.

$$\frac{d\vec{H}}{dt} = \vec{F}_a \quad (1.3)$$

$$\frac{d\vec{b}}{dt} = \vec{Q} \quad (1.4)$$

Tyto rovnice říkají že, že časová změna hybnosti H v inerciální souřadné soustavě je rovna výsledné zrychlující síle F_a působící na letoun a časová změna momentu hybnosti (točivosti) b v inerciální souřadné soustavě je rovna výslednému zrychlujícímu momentu Q působícímu na letoun [1].

Z těchto dvou základních rovnic se lze postupnými úpravami dopracovat k sestavení šesti pohybových rovnic letounu, jejichž maticovou formu představují rovnice 1.5 a 1.6 [1].

$$m \begin{Bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{Bmatrix} + m \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} + mg \begin{Bmatrix} -\sin\Theta \\ \cos\Theta \sin\Phi \\ \cos\Theta \cos\Phi \end{Bmatrix} \quad (1.5)$$

$$\begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{yx} & I_y & -I_{yz} \\ -I_{zx} & -I_{zy} & I_z \end{bmatrix} \begin{Bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix} + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{yx} & I_y & -I_{yz} \\ -I_{zx} & -I_{zy} & I_z \end{bmatrix} \begin{Bmatrix} p \\ q \\ r \end{Bmatrix} = \begin{Bmatrix} L \\ M \\ N \end{Bmatrix} \quad (1.6)$$

Tento systém se ještě doplňuje o kinematické vazby, které dávají do souvislosti časové derivace polohových Eulerových úhlů a úhlové rychlosti v letadlové souřadné soustavě 1.7.

$$\begin{Bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{Bmatrix} = \begin{bmatrix} 1 & \sin\Phi \cdot \tan\Theta & \cos\Phi \cdot \tan\Theta \\ 0 & \cos\Phi & -\sin\Phi \\ 0 & \sin\Phi/\cos\Theta & \cos\Phi/\cos\Theta \end{bmatrix} \begin{Bmatrix} p \\ q \\ r \end{Bmatrix} \quad (1.7)$$

Vzhledem k tomu, že letouny bývají zpravidla souměrné k rovině (x,y), lze považovat deviační momenty $I_{x,y}$ a $I_{y,z}$ za nulové. Inerční matice vystupující v rovnici 1.6 se proto zjednodušuje na do tvaru 1.8 [1].

$$[I] = \begin{bmatrix} I_x & 0 & -I_{xz} \\ 0 & I_y & 0 \\ -I_{zx} & 0 & I_z \end{bmatrix} \quad (1.8)$$

Uvedené soustava rovnic vystihuje pohyb letounu jako hmotného bodu, pro účely spojené se zkoumáním dynamické stability ale není příliš vhodná. Proto se zavádějí takzvané odchylkové pohybové rovnice ve zmiňované stabilitní souřadnicové soustavě, které vycházejí z rovnic 1.5 a 1.6. Odchylkové rovnice umožňují pracovat pouze s odchylkami od referenčních hodnot, což značně zjednodušuje popis pohybu letounu.

Dalšího zjednodušení se dosahuje uvažováním malých odchylek, čímž se zredukuje výskyt goniometrických funkcí a mocnin v rovnicích. Nelinaritu, kterou do soustavy vnáší síly a momenty na pravých stranách se lze zbavit pomocí linearizace v podobě rozvinutí Taylorovy řady v oblasti referenčního bodu v závislosti na poruchových veličinách a uvažováním pouze prvních členů.

Soustava se dále rozšiřuje o kinematické vazby 1.7 Díky symetrii letounu lze soustavu rozdělit na dvě nezávislé soustavy reprezentující podélný 1.9 a stranový 1.10 pohyb letounu [1].

$$\begin{Bmatrix} \Delta \dot{u} \\ \Delta \dot{w} \\ \Delta \dot{q} \\ \Delta \dot{\Theta} \end{Bmatrix} = \begin{bmatrix} x_u & x_w & 0 & x_\Theta \\ z_u & z_w & z_q & z_\Theta \\ m_u & m_w & m_q & m_\Theta \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{Bmatrix} \Delta u \\ \Delta w \\ \Delta q \\ \Delta \Theta \end{Bmatrix} + \begin{bmatrix} 0 & 0 \\ z_{\delta v} & 0 \\ m_{\delta v} & m_{\dot{\delta v}} \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \Delta \delta_v \\ \Delta \dot{\delta}_v \end{Bmatrix} \quad (1.9)$$

Tab. 1.1: Členy soustavy 1.9

index ↓	x	z	m
u	$\frac{X_u}{m}$	$\frac{Z_u}{m-Z_{\dot{w}}}$	$\frac{1}{I_y} \left(M_u + \frac{M_{\dot{w}} Z_u}{m-Z_{\dot{w}}} \right)$
w	$\frac{X_w}{m}$	$\frac{Z_w}{m-Z_{\dot{w}}}$	$\frac{1}{I_y} \left(M_w + \frac{M_{\dot{w}} Z_w}{m-Z_{\dot{w}}} \right)$
q	0	$\frac{m \cdot u_R + Z_q}{m-Z_{\dot{w}}}$	$\frac{1}{I_y} \left(M_q + \frac{M_{\dot{w}} (m u_R + Z_q)}{m-Z_{\dot{w}}} \right)$
Θ	$-g \cdot \cos \Theta_R$	$\frac{m \cdot g \cdot \sin \Theta_R}{m-Z_{\dot{w}}}$	$-\frac{M_{\dot{w}} \cdot m \cdot g \sin \Theta_R}{I_y (m-Z_{\dot{w}})}$
δ_v	0	$\frac{Z_{\delta v}}{m-Z_{\dot{w}}}$	$\frac{1}{I_y} \left(M_{\delta v} + \frac{M_{\dot{w}} \cdot Z_{\delta v}}{m-Z_{\dot{w}}} \right)$
$\dot{\delta}_v$	0	0	$\frac{M_{\dot{\delta}_v}}{I_y}$

$$\begin{pmatrix} \Delta \dot{v} \\ \Delta \dot{p} \\ \Delta \dot{r} \\ \Delta \dot{\Phi} \end{pmatrix} = \begin{bmatrix} y_v & y_p & y_r & y_\Phi \\ l_v & l_p & l_r & 0 \\ n_v & n_p & n_r & 0 \\ 0 & 1 & k_r & 0 \end{bmatrix} \begin{pmatrix} \Delta v \\ \Delta p \\ \Delta r \\ \Delta \Phi \end{pmatrix} + \begin{bmatrix} 0 & 0 & z_{\delta s} & 0 \\ l_{\delta k} & l_{\dot{\delta} k} & l_{\delta s} & l_{\dot{\delta} s} \\ n_{\delta k} & n_{\dot{\delta} k} & n_{\delta s} & n_{\dot{\delta} s} \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} \Delta \delta_k \\ \Delta \dot{\delta}_k \\ \Delta \delta_s \\ \Delta \dot{\delta}_s \end{pmatrix} \quad (1.10)$$

Tab. 1.2: Členy soustavy 1.10

index ↓	y	l	n	k
v	$\frac{Y_v}{m}$	$\frac{I_z L_v + I_{xz} N_v}{I_x I_z - I_{xz}^2}$	$\frac{I_{xz} L_v + I_x N_v}{I_x I_z - I_{xz}^2}$	0
p	$\frac{Y_p}{m}$	$\frac{I_z L_p + I_{xz} N_p}{I_x I_z - I_{xz}^2}$	$\frac{I_{xz} L_p + I_x N_p}{I_x I_z - I_{xz}^2}$	1
r	$\frac{Y_r}{m} - u_R$	$\frac{I_z L_r + I_{xz} N_r}{I_x I_z - I_{xz}^2}$	$\frac{I_{xz} L_r + I_x N_r}{I_x I_z - I_{xz}^2}$	$tg \Theta_R$
Φ	$g \cos \Theta_R$	0	0	0
δ_k	0	$\frac{I_z L_{\delta k} + I_{xz} N_{\delta k}}{I_x I_z - I_{xz}^2}$	$\frac{I_{xz} L_{\delta k} + I_x N_{\delta k}}{I_x I_z - I_{xz}^2}$	0
$\dot{\delta}_k$	0	$\frac{I_z L_{\dot{\delta} k}}{I_x I_z - I_{xz}^2}$	$\frac{I_{xz} L_{\dot{\delta} k}}{I_x I_z - I_{xz}^2}$	0
δ_s	$\frac{Y_{\delta s}}{m}$	$\frac{I_z L_{\delta s} + I_{xz} N_{\delta s}}{I_x I_z - I_{xz}^2}$	$\frac{I_{xz} L_{\delta s} + I_x N_{\delta s}}{I_x I_z - I_{xz}^2}$	0
$\dot{\delta}_s$	0	$\frac{I_{xz} L_{\dot{\delta} s}}{I_x I_z - I_{xz}^2}$	$\frac{I_x N_{\dot{\delta} s}}{I_x I_z - I_{xz}^2}$	0

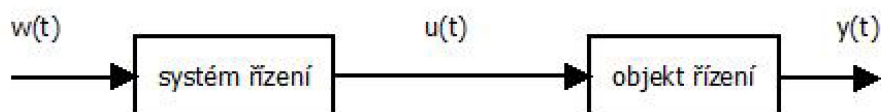
Soustavy jsou uvedené v takzvaném normálním tvaru, jejich členy obsahující rozměrové stabilitní a říditelnostní derivace jsou uvedeny v tabulkách 1.1 a 1.2. Odvození pohybových rovnic lze najít například v [1].

1.2 Letoun jako dynamický systém

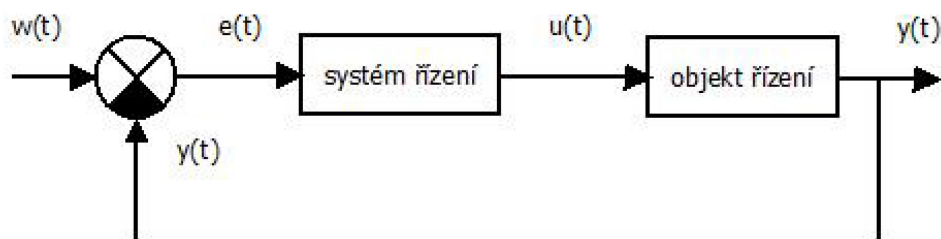
Při zkoumání chování fyzikálních dějů pomocí matematických aparátů by bylo velmi obtížné obsáhnout veškerou realitu. Z tohoto důvodu se zavádí pojem systém. Systém je izolovaný model, obsahující právě tu část objektivní reality, která je středem zkoumání. Tato část se nazývá "objekt" a vše ostatní "okolí". V případě, popisovaném v této práci, se za objekt považuje letoun, případně části letounu ovlivňující podélný, nebo stranový pohyb, pokud je předmětem zkoumání jen jeden z nich [7].

Dynamický systém je takový, který se nějakým způsobem vyvíjí v čase. Obecně se rozděluje na neřízený, kdy jde o uzavřený systém bez vstupů a řízený, jehož výstup $y(t)$ je ovlivňován vstupní charakteristikou $u(t)$ a její historií. Pokud existuje zpětná vazba vstupu na výstup, pak se jedná o regulovaný dynamický systém. V opačném případě jde o řízení, které se nazývá ovládání. U regulovaných dynamických systémů se pak vstupní signál upravuje na základě výstupního tak, aby výstupní děj dosáhl předem požadovaných hodnot [7, 1].

a)



b)



Obr. 1.4: Systém řízení a) a ovládání b)

Při řešení úloh spojených s mechanikou letu můžeme považovat letoun za dynamický systém a popisovat jeho chování metodami používanými v teorii řízení, jako je například popis letounu ve stavovém prostoru, jehož obecný zápis je tvořen rovnicemi 1.11 a 1.12.

$$\dot{x}(t) = \mathbf{A}x(t) + \mathbf{B}u(t) \quad (1.11)$$

$$\dot{y}(t) = \mathbf{C}x(t) + \mathbf{D}u(t) \quad (1.12)$$

Výpočtový model pro pohyb letounu ve stavovém prostoru můžeme vyjádřit přímo pomocí maticového zápisu pohybových rovnic 1.9 a 1.10.

2 MATLAB

Matlab je vysokoúrovňový skriptovací programovací jazyk a interaktivní prostředí, užívané pro inženýrské výpočty a vizualizaci dat po celém světě. Nabízí širokou škálu možností ke zpracování a vizualizaci dat, tvorbu algoritmů, simulací a vytváření aplikací, což z něho spolu s příjemným uživatelským prostředím dělá užitečný nástroj v inženýrské praxi [8, 6].

2.1 Programovací prostředí Matlab a Simulink

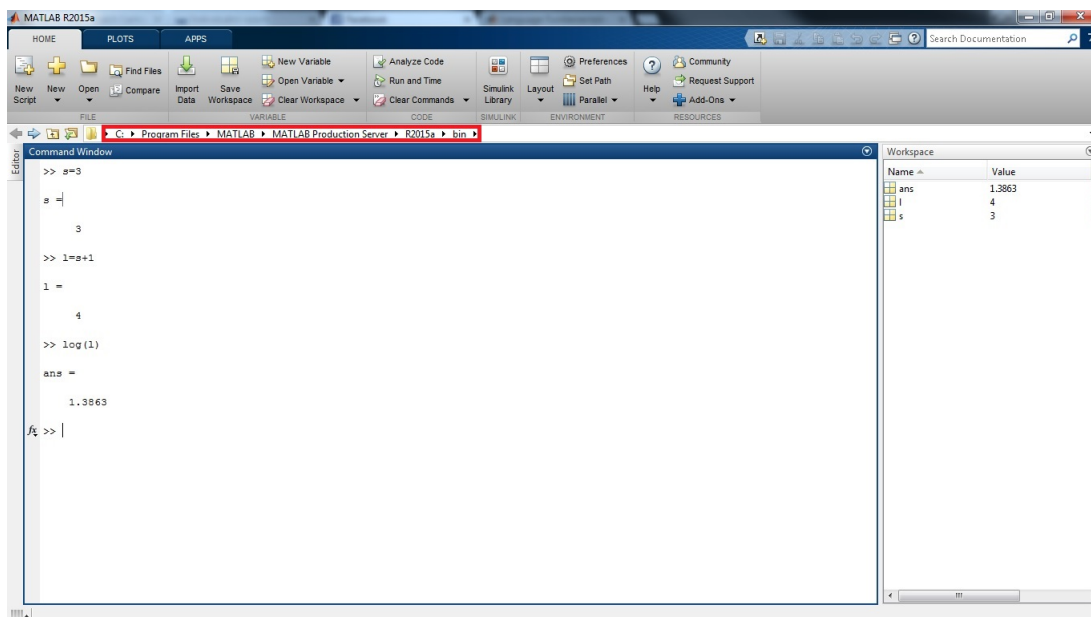
Matlab je produktem společnosti MathWorks. Základním datovým prvkem v tomto programu je matice, což umožňuje rychlé řešení mnoha numerických problémů. Velmi ocenitelná vlastnost Matlabu je jeho snadná rozšiřitelnost umožňující doplňovat systém o uživatelem vytvořené funkce, nebo celé aplikace. Navíc lze získat celá řada nadstaveb (toolboxů), což je soubor funkcí pro řešení určitých tříd problémů [6].

Součástí Matlabu je Simulink. Jde o rozšíření, díky kterému lze vytvářet funkce a simulace přehledným způsobem za pomoci blokových schémat [8].

Pro účely zpracování této práce byla použita verze Matlab R2015a.

2.1.1 Uživatelské prostředí Matlabu

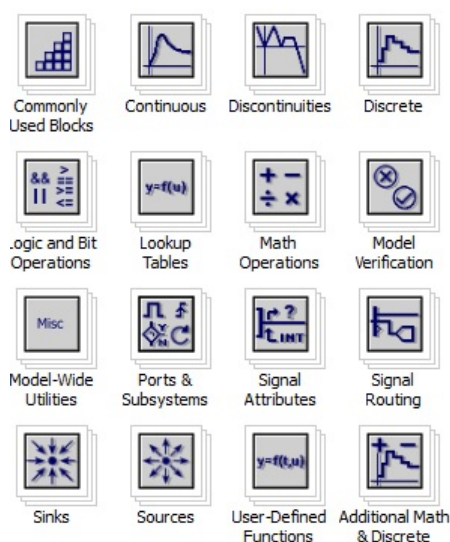
Pracovní prostředí použité verze Matlabu je situováno do tří základních částí: command window určené k zadávání jednoduchých příkazů, workspace obsahující proměnné a výstupy funkcí volaných v command window a editor, který slouží k vytváření skriptů, funkcí, tříd - tedy m-filů. Důležité je před použitím programu zkontrolovat vyhledávací cestu (path). Ta je vyznačená červeně na obr. 2.1 a určuje složku, ve které Matlab hledá volané funkce definované uživatelem. Tuto cestu je třeba nastavit, pokud chceme použít m-file uložený mimo místa, ke kterým má Matlab přístup. Ty lze zobrazit příkazem *path*. Pokud je potřeba vyhledávat z více nepřednastavených míst, lze rozšířit seznam cest pomocí příkazu *addpath*, ovšem za cenu mírného zpomalení programu.



Obr. 2.1: Uživatelské rozhraní Matlabu R2015a

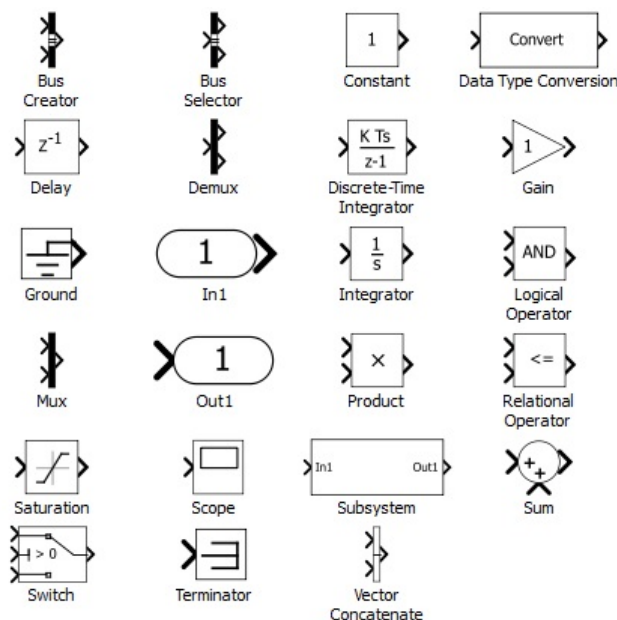
2.1.2 Matlab simulink

Simulink je jedno z nejpoužívanějších rozšíření Matlabu. Využívá Matlab pro modelování, simulaci a analýzu dynamických systémů v přehledném grafickém uživatelském prostředí pomocí propojování bloků reprezentujících funkce [5].



Obr. 2.2: Skupiny bloků v základní knihovně Simulinku

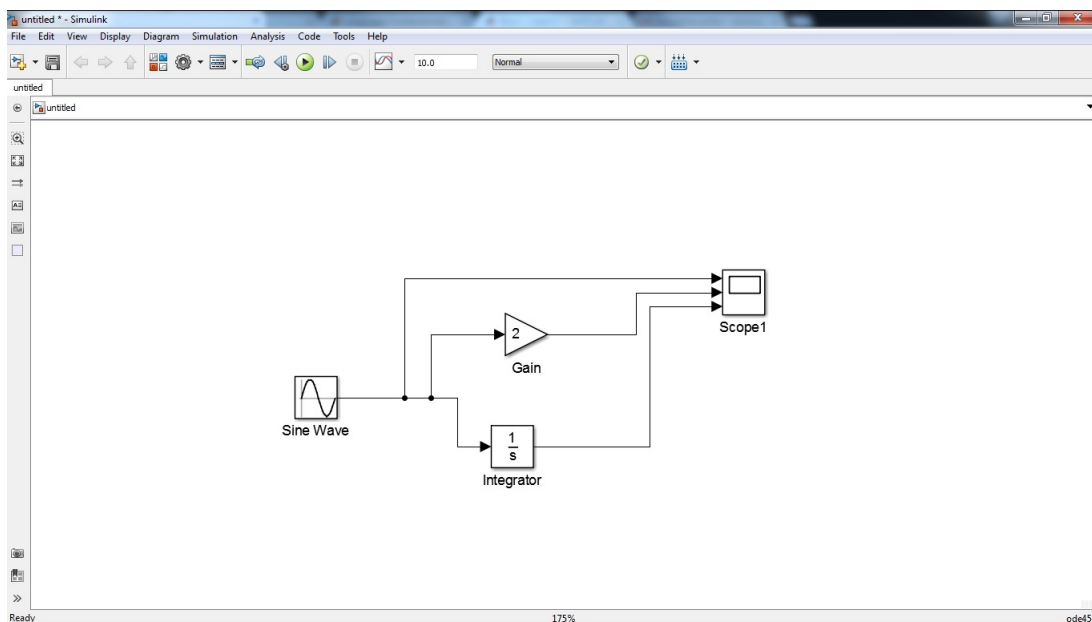
Simulink obsahuje knihovnu bloků pro návrh modelů rozdělenou do šestnácti skupin zobrazených na obrázku 2.2. Nejpoužívanější bloky zařazené do skupiny Commonly Used Blocks jsou na obrázku 2.3. Návrh nového systému spočívá v přesouvání bloků z knihovny do pracovního prostoru a jejich propojování šipkami, naznačujícími směr toku dat.



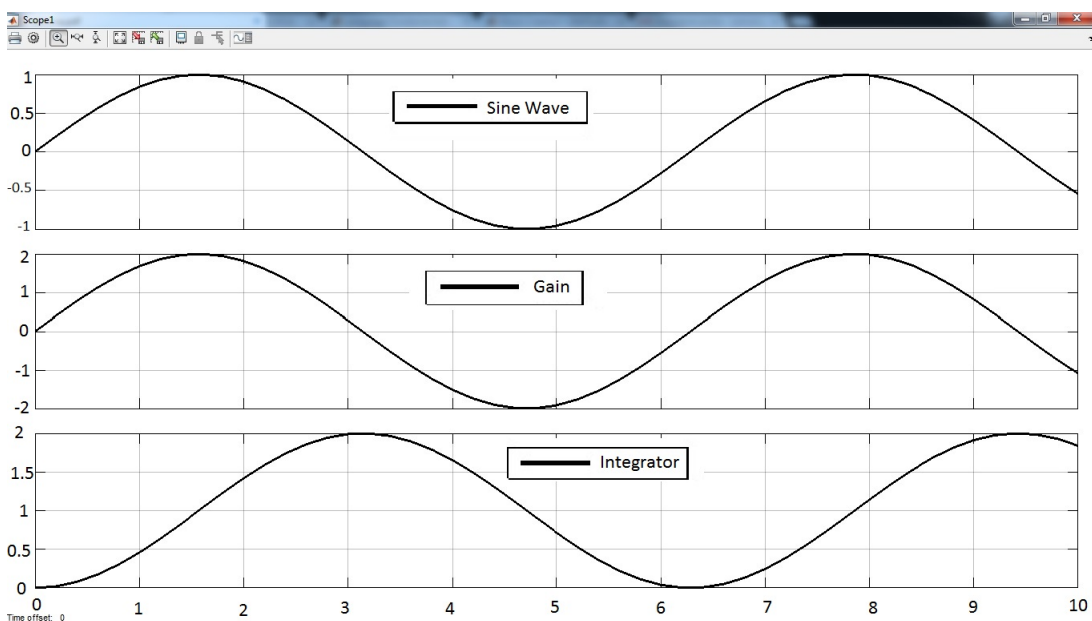
Obr. 2.3: Nejčastěji používané bloky

Jednoduchý blokový daigram v Simulinku je na obrázku 2.4. Zdrojový signál generuje blok *Sine Wave* jako sinusovou křivku. Ve směru šipek pokračují data do bloků *Gain*, který násobí signál a *Integrator*, který ho integruje. Zdrojový, znásobený a integrovaný signál jsou pak přivedeny do bloku *Scope1*, který má schopnost zobrazovat přivedené signály v grafu. Rozkliknutím daného bloku lze nastavit jeho parametry. Pro amplitudu sinusového signálu 1, frekvenci 1 rad/s a hodnotu násobiče *Gain* 2 je výstup bloku *Scope1* po spuštění desetisekundové simulace zobrazen na obrázku 2.5.

Knihovnu Simulinku lze rozšířit zakoupením toolboxů, nebo vytvářením vlastních funkčních bloků.



Obr. 2.4: Blokový model v Simulinku

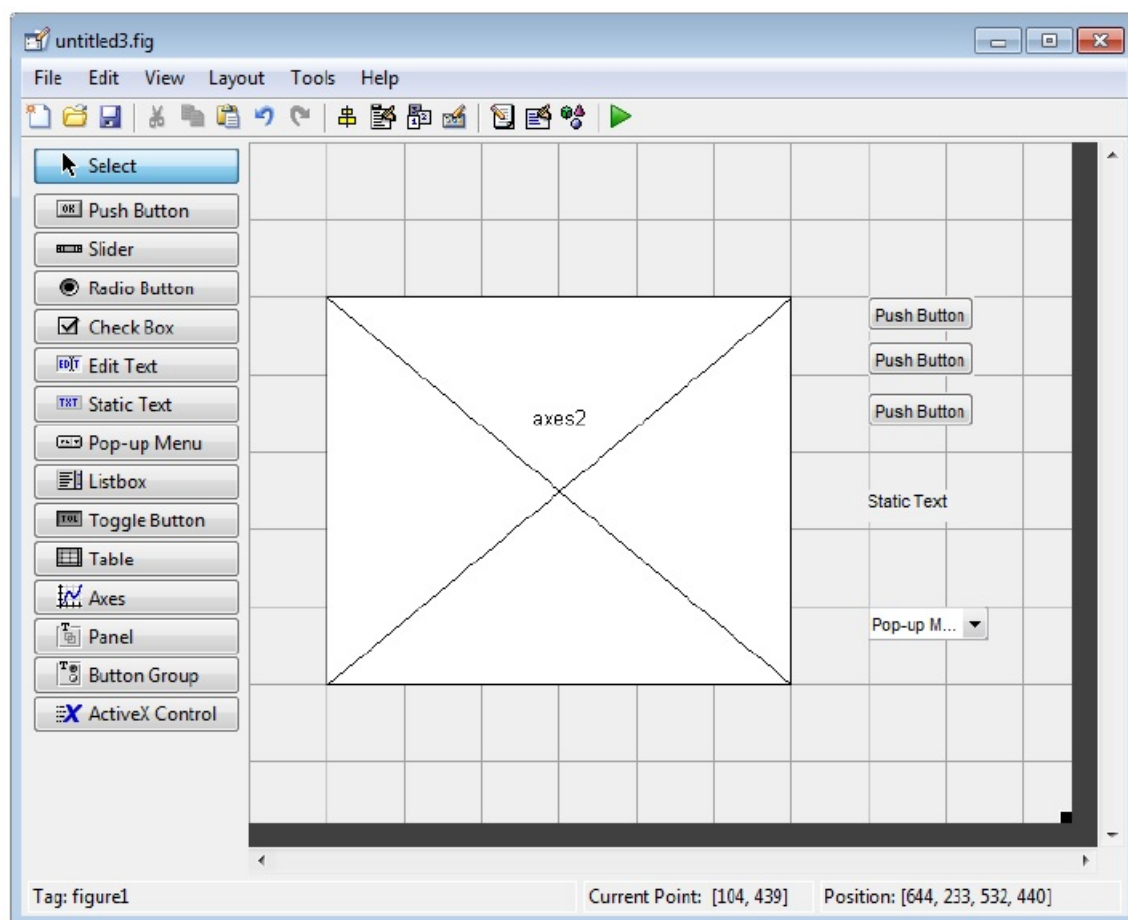


Obr. 2.5: Výstup z blokového modelu na obr. 2.4

2.1.3 Matlab guide

Matlab podporuje návrh grafického uživatelského rozhraní (dále UI). Existují dva způsoby jak UI navrhnu - napsáním kódu, ve kterém jsou definované všechny prvky, jejich vlastnosti a chování, nebo pomocí GUIDE. GUIDE je zabudované prostředí

v Matlabu, které umožňuje vytvoření a nastavení prvků UI. Je to nástroj, který zjednodušuje a zpřehledňuje tvorbu jednodušších UI [10].



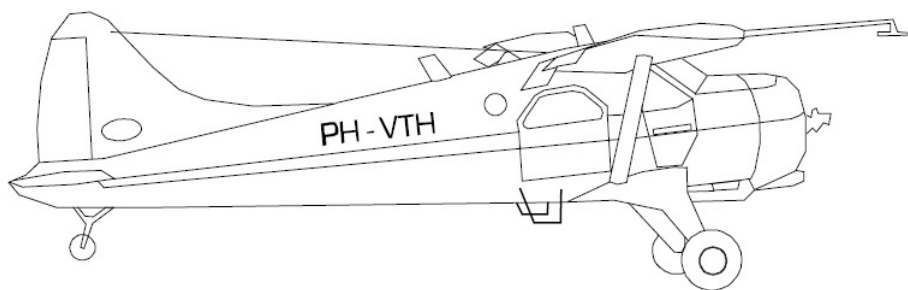
Obr. 2.6: Prostředí pro návrh UI [10]

Guide lze vyvolat rozbalením ikony New z palety nabídek a vybráním Graphical User Interface, nebo příkazem *guide* napsaným do Command Window. Ve vyvolaném okně se nabízí na výběr několik předdefinovaných UI, které můžou sloužit jako vzory a Blank GUI, který nemá předdefinované žádné funkce. Na obrázku 2.6 je Blank GUI s nastavenými některými funkcemi z panelu na levé straně okna.

Podrobný návod jak vytvářet uživatelská rozhraní v Matlabu je v literatuře [10].

3 FDC

FDC je zkratka pro anglický název *Flight Dynamic and Control*. FDC Blockset pro Matlab a Simulink je volně šiřitelné rozšíření Matlabu, umožňující analyzovat dynamiku letounu a řídicí systémy. Autorem je Marc Rauw, který navrhl nelineární model pro letoun "Beaver", na základě něhož vznikl tento toolbox. Knihovna toolboxu obsahuje kromě nelineárního modelu letounu, jehož vnitřní struktura je na obrázku 3.2 model atmosférických poruch, modely radiové navigace letounu, autopilota situovaného pro letoun "Beaver" a několik užitečných funkcí zjednodušujících práci se systémem [9].



Obr. 3.1: Letoun Beaver, pro který byl původně navržen dynamický model [9]

3.1 Teoretické pozadí nelineárního modelu

V nelineárním modelu navrženém v FDC je letoun považovaný za dokonale tuhý model jehož pohyb lze v souladu s Newtonovou základní mechanikou popsat pomocí silové a momentové rovnice v následujícím tvaru [9].

$$\mathbf{F} = m \left(\frac{\partial \mathbf{V}}{\partial t} + \boldsymbol{\Omega} \times \mathbf{V} \right) \quad (3.1)$$

$$\mathbf{M} = \frac{\partial (\mathbf{I} \cdot \boldsymbol{\Omega})}{\partial t} + \boldsymbol{\Omega} + (\mathbf{I} \cdot \boldsymbol{\Omega}) \quad (3.2)$$

Kde m je hmotnost letounu, \mathbf{V} je vektor rychlostí těžiště v letadlové souřadné soustavě $[u, v, w]^T$, $\boldsymbol{\Omega}$ je vektor úhlových rychlostí $[p, q, r]^T$, \mathbf{F} je úplný vektor

vnějších sil $[F_x, F_y, F_z]^T$ a \mathbf{M} je úplný vektor vnějších momentů $[L, M, N]^T$. \mathbf{I} je tensor setrvačnosti, který je definován jako:

$$\mathbf{I} = \begin{bmatrix} I_{xx} & -J_{xy} & -J_{xz} \\ -J_{yx} & I_{yy} & -J_{yz} \\ -J_{zx} & -J_{zy} & I_{zz} \end{bmatrix} \quad (3.3)$$

Jehož koeficienty jsou momenty setrvačnosti a deviační momenty k příslušným plochám. Přepsáním rovnic 3.1 a 3.2 do diferenciálního tvaru 3.4 a 3.5 získáme použitelné vztahy pro vytvoření dynamického systému.

$$\frac{\partial \mathbf{V}}{\partial t} = \frac{\mathbf{F}}{m} - \Omega \times \mathbf{V} \quad (3.4)$$

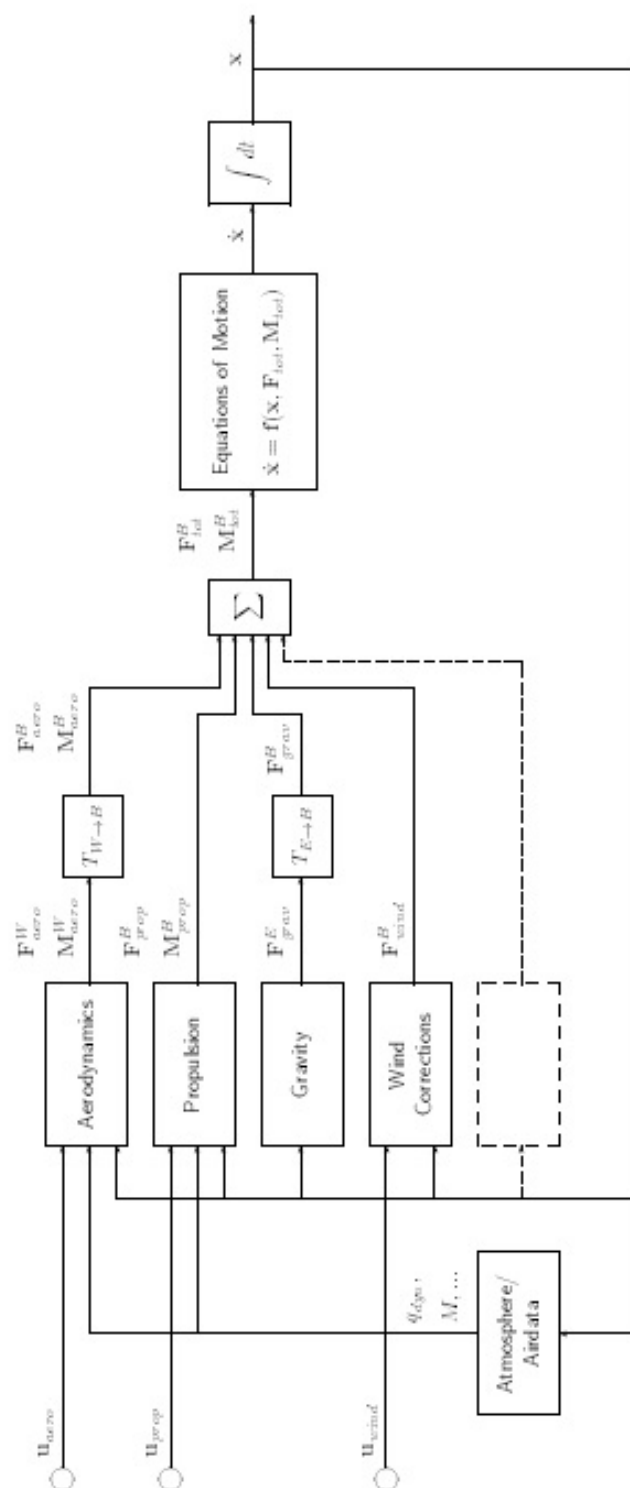
$$\frac{\partial \Omega}{\partial t} = \mathbf{I}^{-1} (\mathbf{M} - \Omega \times \mathbf{I} \cdot \Omega) \quad (3.5)$$

Tyto rovnice společně tvoří universální stavový prostor pro jakékoli tuhé těleso a zároveň jsou základem popisovaného dynamického modelu. Vstup do tohoto systému tedy tvoří silové a momentové vnější působení a výstupem jsou rychlosti a zrychlení tělesa. Jednoduchost této představy ovšem narušuje skutečnost, že velikost sil a momentů působících na konstrukci je závislá na stavových veličinách, mezi které se řadí i rychlosti a zrychlení. Jinými slovy síly a momenty které způsobují změnu kinematických hodnot jsou závislé na aktuálním kinematickém stavu, ve kterém se v našem případě letoun nachází. Přestože tím roste komplexnost modelu, není tím znemožněna jeho realizace [9].

Rauw ve svém modelu vytvořil takzvaný stavový vektor

$$x = [V, \alpha, \beta, p, q, r, \psi, \theta, \varphi, x_e, y_e, H]^T \quad (3.6)$$

kde V je pravá vzdušná rychlost, α úhel náběhu, β úhel vybočení, p, q a r jsou úhlové rychlosti v letadlové souřadné soustavě, ψ, θ a φ jsou úhly natočení v zemské souřadné soustavě, x_e a y_e jsou souřadnice polohy letounu a H je jeho výška nad mořem. Stavový vektor využívá pro vnitřní výpočty ve svém modelu a je nutné ho před spuštěním simulace definovat jakožto vektor počátečních podmínek. Stavový vektor je zároveň výstupem modelu. Na obrázku 3.2 si lze povšimnout zpětné vazby, která zaručuje že v dalším kroce výpočtu bude působení sil na letoun vypočítáno na základě stavových podmínek v předchozím kroku. Tím je vytvořena vnitřní smyčka dávající do souvislosti stavové hodnoty a vstupní hodnoty, což elegantně řeší problém načrtnutý v předchozím odstavci [9].



Obr. 3.2: Blokový diagram struktury FDC [9]

3.1.1 Vnější síly a momenty

Dynamický model rozděluje vnější působení do několika skupin: Aerodynamické síly a momenty, silové působení od propulze, gravitační sílu a vliv nestálé atmosféry. V diagramu 3.2 je čárkovane naznačený pátý blok, který reprezentuje síly od pojiždění a vliv blízkosti země při letu. Tyto nejsou do modelu FDC zahrnuty, ale lze je v případě potřeby doplnit [9].

Pro výpočet aerodynamických sil a momentů je třeba znát aktuální stav stavového vektoru x a okamžité výchylky řídicích ploch. Kvantifikace jejich hodnot je realizována na základě polynomických funkcí ve tvaru 3.7.

$$F_{aero} = d \cdot p_1 \cdot (x; \dot{x}; u_{aero}) \quad [9] \quad (3.7)$$

Kde F_{aero} je vektor aerodynamických sil a momentů, p_1 je vektor polynomů obsahujících bezrozměrné silové a momentové derivace. Ten je zobrazený na obrázku 3.3. Časová derivace stavového vektoru x v tomto případě obsahuje pouze závislost stranové síly Y_a na změně úhlu vybočení v čase $\dot{\beta}$.

$$\begin{aligned} C_{X_a} &= C_{X_0} + C_{X_\alpha} \alpha + C_{X_{\alpha^2}} \alpha^2 + C_{X_{\alpha^3}} \alpha^3 + C_{X_q} \frac{q\bar{c}}{V} + C_{X_{\delta_r}} \delta_r + C_{X_{\delta_f}} \delta_f + C_{X_{\alpha\delta_f}} \alpha \delta_f \\ C_{Y_a} &= C_{Y_0} + C_{Y_\beta} \beta + C_{Y_p} \frac{pb}{2V} + C_{Y_r} \frac{rb}{2V} + C_{Y_{\delta_a}} \delta_a + C_{Y_{\delta_r}} \delta_r + C_{Y_{\delta_r\alpha}} \delta_r \alpha + C_{Y_{\dot{\beta}}} \frac{\dot{\beta}b}{2V} \\ C_{Z_a} &= C_{Z_0} + C_{Z_\alpha} \alpha + C_{Z_{\alpha^3}} \alpha^3 + C_{Z_q} \frac{q\bar{c}}{V} + C_{Z_{\delta_e}} \delta_e + C_{Z_{\delta_e\beta^2}} \delta_e \beta^2 + C_{Z_{\delta_f}} \delta_f + C_{Z_{\alpha\delta_f}} \alpha \delta_f \\ C_{l_a} &= C_{l_0} + C_{l_\beta} \beta + C_{l_p} \frac{pb}{2V} + C_{l_r} \frac{rb}{2V} + C_{l_{\delta_a}} \delta_a + C_{l_{\delta_r}} \delta_r + C_{l_{\delta_a\alpha}} \delta_a \alpha \\ C_{m_a} &= C_{m_0} + C_{m_\alpha} \alpha + C_{m_{\alpha^2}} \alpha^2 + C_{m_q} \frac{q\bar{c}}{V} + C_{m_{\delta_e}} \delta_e + C_{m_{\beta^2}} \beta^2 + C_{m_r} \frac{rb}{2V} + C_{m_{\delta_f}} \delta_f \\ C_{n_a} &= C_{n_0} + C_{n_\beta} \beta + C_{n_p} \frac{pb}{2V} + C_{n_r} \frac{rb}{2V} + C_{n_{\delta_a}} \delta_a + C_{n_{\delta_r}} \delta_r + C_{n_q} \frac{q\bar{c}}{V} + C_{n_{\beta^3}} \beta^3 \end{aligned}$$

Obr. 3.3: Vektor polynomických funkcí pro výpočet sil a momentů [9]

Vektor d je ve tvaru 3.8.

$$d = q_{dyn} \cdot S \cdot \left[1, 1, 1, \frac{b}{2}, \bar{c}, \frac{b}{2} \right] \quad [9] \quad (3.8)$$

Kde $q_{dyn} = \frac{1}{2} \cdot \rho \cdot v^2$ je dynamický tlak, S je plocha křídla a \bar{c} je střední aerodynamická tětiva.

Síly a momenty od propulze silně závisí na typu letounu. V modelu FDC je výpočet uskutečňován obdobným způsobem jako výpočet aerodynamických sil - pomocí polynomických funkcí obsahujících bezrozměrné derivace 3.9.

$$F_{prop} = d \cdot p_2 \cdot (x; dpt) \quad [9] \quad (3.9)$$

Kde p_2 je vektor polynomů 3.4 závislých na stavovém vektoru x a hodnotě dpt , vypočítávané dle empirického vztahu 3.10 a 3.11.

$$P = 0.7355 \left\{ -326.5 + \left(0.00412 * (p_z + 7.4) (n + 2010) + (408 - 0.0965n) \left(1 - \frac{\rho}{\rho_0} \right) \right) \right\} \quad (3.10)$$

$$dpt = C_1 + C_2 \left(\frac{P}{\frac{1}{2}\rho V^3} \right) \quad (3.11)$$

Kde C_1 a C_2 jsou konstanty závislé na typu letounu, n jsou otáčky vrtule, ρ je hustota ve výšce letu a ρ_0 je hustota vzduchu u země.

$$\begin{aligned} C_{X_p} &= C_{X_{dpt}} dpt + C_{X_{\alpha dpt^2}} \alpha dpt^2 \\ C_{Y_p} &= 0 \\ C_{Z_p} &= C_{Z_{dpt}} dpt \\ C_{l_p} &= C_{l_{\alpha^2 dpt}} \alpha^2 dpt \\ C_{m_p} &= C_{m_{dpt}} dpt \\ C_{n_p} &= C_{n_{dpt^3}} dpt^3 \end{aligned}$$

Obr. 3.4: Vektor polynomických funkcí pro výpočet sil a momentů od propulze [9]

Gravitační sílu v letadlové souřadné soustavě model počítá dle 3.12.

$$F_{grav} = \begin{bmatrix} X_{grav} \\ Y_{grav} \\ Z_{grav} \end{bmatrix} = W \cdot \begin{bmatrix} -\sin\theta \\ \cos\theta \sin\varphi \\ \cos\theta \cos\varphi \end{bmatrix} \quad (3.12)$$

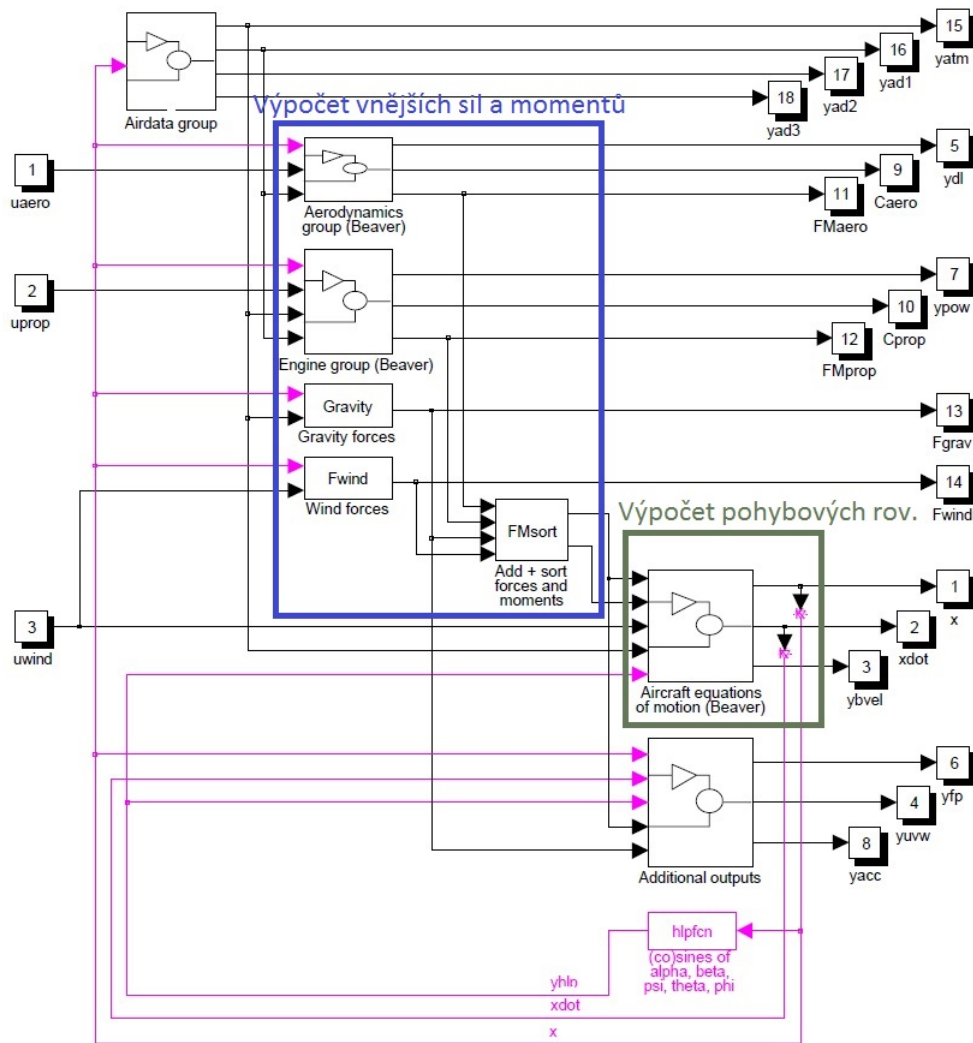
Kde $W = m \cdot g$

Vliv nestabilní atmosféry a jeho výpočet je podrobně popsán v [9]. V modelu upraveném pro účely této práce zůstal zachovaný, ale není využitý.

3.2 Implementace nelineárního modelu FDC

Na obrázku 3.5 je zobrazen blokový diagram vytvořený v prostředí Matlab Simulink. Jednotlivé části jsou uživatelsky definované funkce a mají vlastní strukturu tvořenou dalšími funkčními bloky. Například základní struktura funkce s názvem *Gravity* je na obrázku 3.6.

Do bloku přicházejí data vstupními porty 1 a 2. Vstupním portem 1 je posílán stavový vektor x 3.6 a portem 2 vektor y_{atm} , který je výstupem bloku s názvem



Obr. 3.5: Implementace dynamického modelu FDC. Upraveno [9]

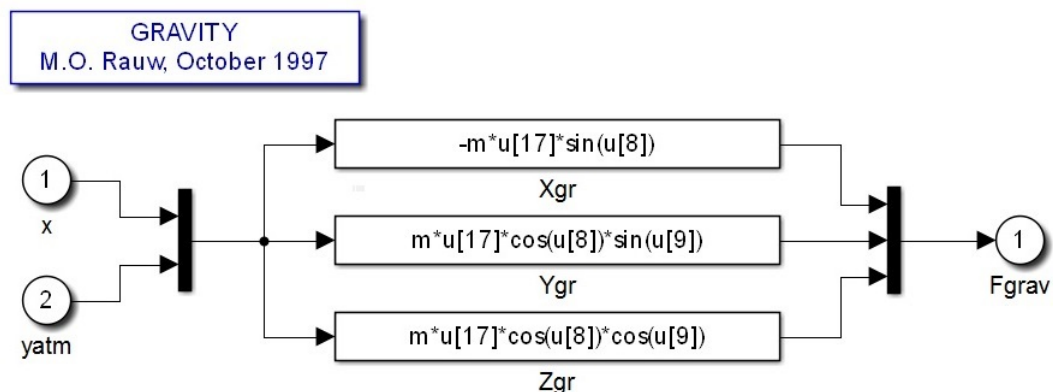
Airdata group a který reprezentuje základní atmosférické vlastnosti jako je hustota vzduchu, statický tlak, teplota, kinematická viskozita a gravitační zrychlení 3.13.

$$y_{atm} = [\rho, p_s, T, \mu, g]^T \quad (3.13)$$

Vstupní vektory jsou spojeny do jednoho vektoru o sedmnácti členech pomocí funkce *mux*. Dále je proveden výpočet na základě rovnice 3.12 použitím *Fcn*, spojení výsledků do vektorového tvaru blokem *mux* a odesláním do výstupního portu 1.

Součástí struktury některých částí jsou další autorem vytvořené blokové funkce. Vzniká tak ještě třetí a někdy až čtvrtá vrstva. Toto rozčlenění do více jednodušších bloků je z hlediska orientace v takovémto složitém programu velmi příjemné.

Na obrázku 3.5 je modře ohraničená oblast, která má na starosti výpočet vnějších sil působících na letoun ve shodě s rovnicemi uvedenými v kapitole 3.1.1 a zeleně je



Obr. 3.6: Vnitřní struktura bloku Gravity z diagramu 3.5

zvýrazněný blok *Aircraft equations of motion*, který počítá pohyb letounu na základě rovnic 3.7 a 3.9.

Již zmiňovaný blok *Airdata group* jehož vstupem je pouze stavový vektor x 3.6 má na čtyřech výstupech základní proměnné týkající se atmosféry, rychlostí, tlaků a některých podobnostních čísel.

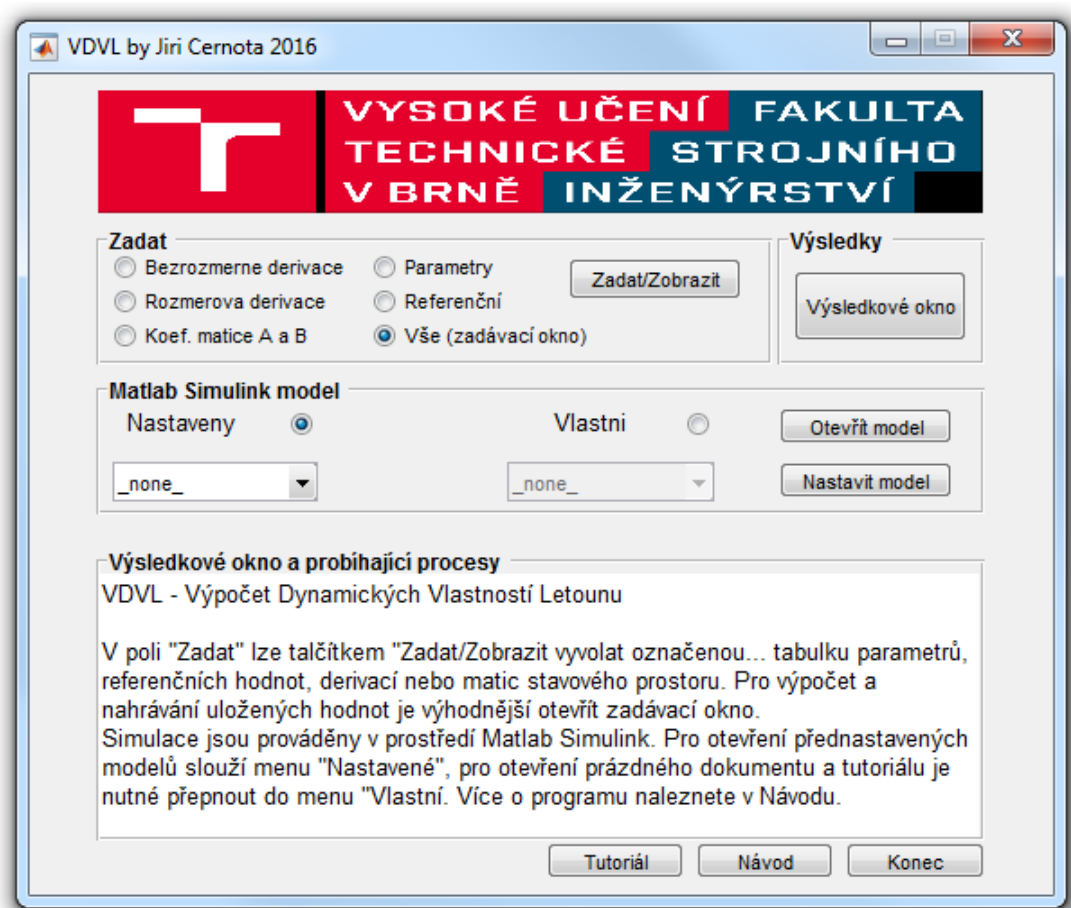
Dalšími částmi modelu jsou bloky *Additional outputs*, který počítá některé proměnné, které nejsou potřebné pro vnitřní výpočty a *hlpfcn*, stanovující goniometrické výpočty, které by se jinak při výpočtu často opakovaly a zpomalovaly by tak simulaci.

Podrobný výčet bloků, jejich vstupů a výstupů, vnitřních struktur a funkcí které v modelu plní uvádí autor v online dostupné dokumentaci [9].

4 VDVL

V rámci této diplomové práce bylo pro účely výuky přemětu Mechanika letu II vytvořeno uživatelské prostředí pro výpočet stabilitních a říditelnostních derivací a matic stavového prostoru pro podélný pohyb letounu, pracovně nazvané VDVL - Výpočet Dynamických Vlastností Letounu. Program interperetovaný v Matalbu dále umožňuje otevírání, nastavování a spouštění dynamických modelů vytvořených v prostředí Matlab Simulink a vizualizaci jejich výstupů přímo z uživatelského rozhraní VDVL. Program tak umožňuje provádění simulací a výpočtů bez hlubších znalostí programovacího jazyka Matlab, což ho dělá vhodným nástrojem pro pochopení některých souvislostí vyučovaným ve zmíněném předmětu.

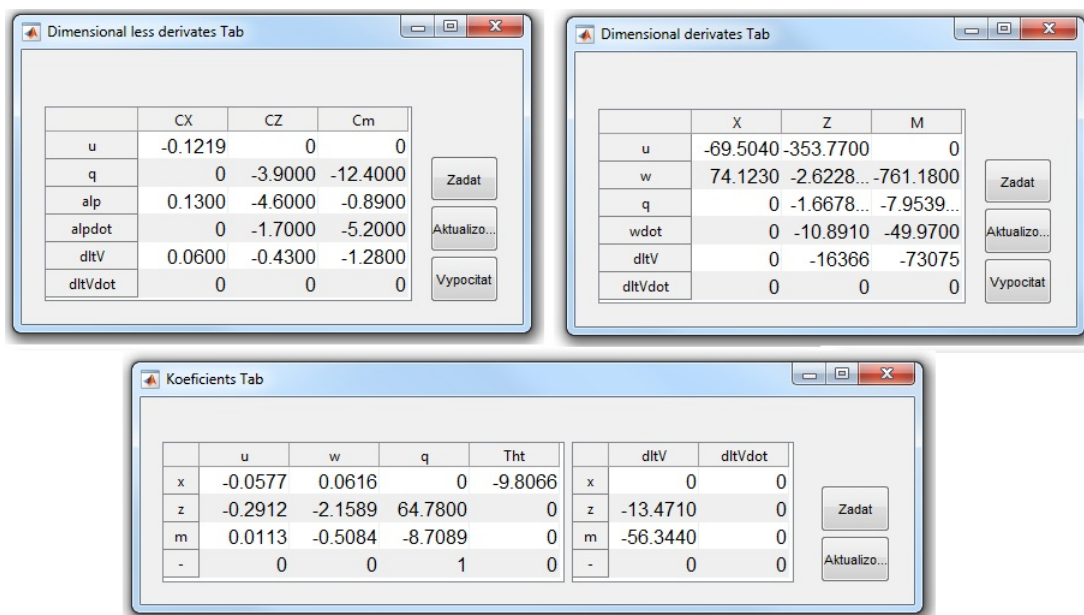
4.1 Základní uživatelské rozhraní



Obr. 4.1: Základní uživatelské rozhraní programu

Na obrázku 4.1 je vzhled základního uživatelského rozhraní programu, které lze vyvolat příkazem *VDVL* v příkazovém řádku Matlabu po nastavení cílové složky na složku obsahující jednotlivé funkce a skripty programu.

Okno programu je rozděleno do čtyř částí - Zadat, Matlab Simulink model, Výsledkové okno a probíhající procesy a Výsledky. V poli "Zadat" lze označení příslušného radiobuttonu otevřít tlačítkem "Zadat/Zobrazit" editovatelné tabulky jednotlivých koeficientů letounu. Součástí těchto tabulek je tlačítko "Zadat", které ukládá aktuální vepsané hodnoty do pracovní paměti počítače. Tabulky Parametrů a derivací pak obsahují tlačítko "vypočítat", které přepočítává zadané hodnoty do tabulek vyšší úrovně (viz 4.5). Tlačítko "aktualizovat" v tabulkách pak nastavuje aktuální hodnoty do dané tabulky, například po provedení výpočtu z nižší úrovně.



Obr. 4.2: Tabulky bezrozměrných derivací, rozměrových derivací a matic stavového prostoru

Dále lze zobrazit Zadávací okno 4.3, které všechny zmíněné tabulky centralizuje do jednoho přehledného grafického výstupu, umožňuje interaktivní zobrazování potřebných hodnot pro výpočet požadovaných hodnot a obsahuje funkci pro ukládání a načítání uložených nastavení z databáze. O zadávání hodnot a výpočtech v programu je pojednáno v kapitole 4.2.

Matice stavového prostoru

Matice soustavy A

	u	w	q	Tht
x	-0.0577	0.0616	0	-9.8066
z	-0.2912	-2.1589	64.7800	0
m	0.0113	-0.5084	-8.7089	0
-	0	0	1	0

Matice řízení B

	dltV	dltVdot
x	0	0
z	-13.4710	0
m	-56.3440	0
-	0	0

Parametry letounu

	Hodnota	
m	1204	[kg]
Iy	1285	[kg*m2]

	Hodnota	
cAK	1.5000	[m]
SK	16.1700	[m2]

Bezrozměrové derivace

	X	Z	M
u	-69.5040	-353.7700	0
w	74.1230	-2.6228	-761.1800
q	0	-1.6678	-7.9539
wdot	0	-10.8910	-49.9700
dltV	0	-16366	-73075
dltVdot	0	0	0

Bezrozměrové derivace

	CX	CZ	Cm
u	-0.1219	0	0
q	0	-3.9000	-12.4000
alp	0.1300	-4.6000	-0.8900
alpdot	0	-1.7000	-5.2000
dltV	0.0600	-0.4300	-1.2800
dltVdot	0	0	0

Referenční hodnoty

	Hodnota	
VR	66.7510	[m/s]
ThtR	0	[rad]
RhoR	1.0565	[kg/m3]
gmmR	0	[rad]

Akce

Zadat
Vypočítat
☒ bezrozměrné derivace
☒ rozměrové derivace
☐ Stavový prostor
 Uložit
 Načíst
 Příklad1
 CC-3
 F4_PHANTOM
 Malý_dopravní_letoun
 Příklad1

POUŽITÍ A NASTAVENÍ ZADÁVACÍHO OKNA
 V pravém horním oknu "Akce" vyberte které hodnoty chcete vypočítat.
 Po zakliknutí se ZELENE vybrání zvolené hodnoty pro výpočet a ORANŽOVÉ hodnoty, které jsou pro výpočet nutné vyplnit
 Výpočet je možné provést pro jakoukoli kombinaci, výpočet je prováděn v pořadí bezrozměrné derivace -> rozměrové derivace -> stavový prostor, s vynecháním nezaškrtnutých hodnot
 Výpočet bude proveden pro:
 - Bezrozměrové derivace
 - Rozměrové derivace

Obr. 4.3: Zadávací okno programu

Část Matlab Simulink model slouží k otevírání přednastavených modelů vytvořených v prostředí Matlab Simulink (radiobutton "Nastaveny"), nového prázdného Simulink souboru a tutoriálu, vytvořeného pro účely výuky přemětu Mechanika letu II. Dále je zde tlačítko "Nastavit model", které odesílá matice z tabulky Koeficienty matice A a B do workspace, ze kterého jsou pak načítané při spouštění simulace modelů pracujících se stavovým prostorem. Tato část programu je podrobněji popsána v kapitole 4.3.

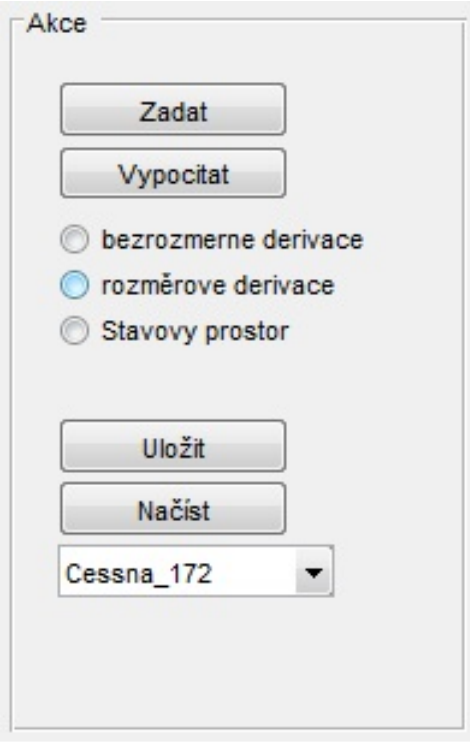
Výsledkové okno a probíhající procesy podává uživateli stručnou nápovědu. Při spouštění grafických objektů a Simulinku, které může trvat i několik sekund, vypisuje informace o spouštění.

Pole Výsledky obsahuje tlačítko pro vyvolání Výsledkového okna. Výsledkové okno slouží k prezentaci výstupů simulací v Simulinku. Více o tomto okně v kapitole 4.4.

4.2 Zadávání a výpočet

Zadání hodnot do programu lze provést po otevření příslušné tabulky 4.2, nebo zadávacího okna 4.3. Při využití programu pro výpočet derivací, případně matic stavového prostoru, je v zadávacím okně pole s názvem "Akce" 4.4, které obsahuje interaktivní prvky pro zpřehlednění výpočtu a ukládání nebo načítání hodnot z nebo

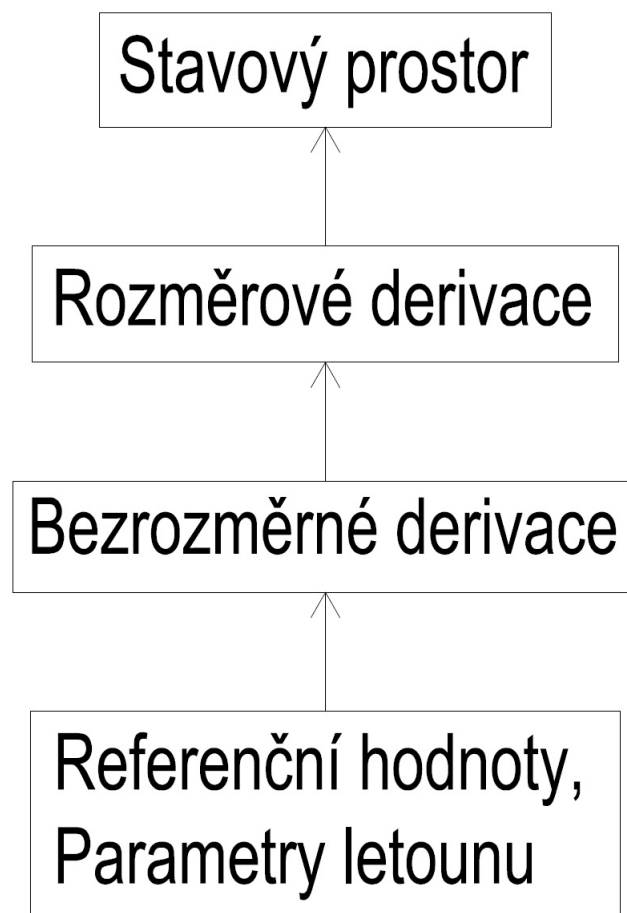
do databáze letounů.



Obr. 4.4: Pole akce v zadávacím okně

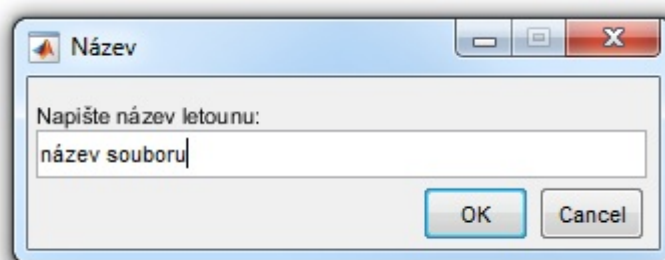
V horní části pole je tlačítko Zadat, které ukládá aktuální hodnoty do pracovní poměti programu. Tlačítko Vypočítat provádí výpočet pro nastavené hodnoty níže umístěnou skupinou radiobuttonů. Označení jednoho, nebo více radiobuttonů se zeleně zbarví buňky tabulek, které program bude počítat a oranžově buňky, které je pro tento výpočet nutné vyplnit (příklad zbarvení je na obrázku 4.3). V případě nevyplnění některé z tázaných hodnot bude výpočet proveden jen pro hodnoty, které danou hodnotu nezohledňují. Závislé parametry se vyhodnotí jako prázdné matice - [].

Při nastavení více hodnot pro výpočet v jenom kroku program postupně provádí výpočet podle postupu na obrázku 4.5. Při přechodu na další úroveň (například z rozměrových derivací na výpočet matic stavového prostoru) využívá hodnoty spočtené v předešlém kroce (pokud byly před zahájením výpočtu ručně vloženy rozměrové derivace a nastavený jejich výpočet, staré hodnoty budou přepsány a pro výpočet matic stavového prostoru se využijí nové spočtené hodnoty). V případě kombinace, která přeskakuje některou z úrovní, se daná tabulka zbarví oranžově a je třeba jí ručně vyplnit (například, pokud je nastavený výpočet pro bezrozměrné derivace a stavový prostor, ale ne rozměrových derivací. Rozměrové derivace pak musí být zadány ručně a pro výpočet stavového prostoru jsou použity zadané hodnoty).



Obr. 4.5: Postup při výpočtu v zadávacím okně

4.2.1 Ukládání a načítání dat z databáze



Obr. 4.6: Dotazovací okno při ukládání zadaných hodnot do databáze

Program VDWL podporuje ukládání zadaných hodnot do databáze a jejich opětovné vyvolání. Uložené hodnoty zůstávají k dispozici i po ukončení programu a lze

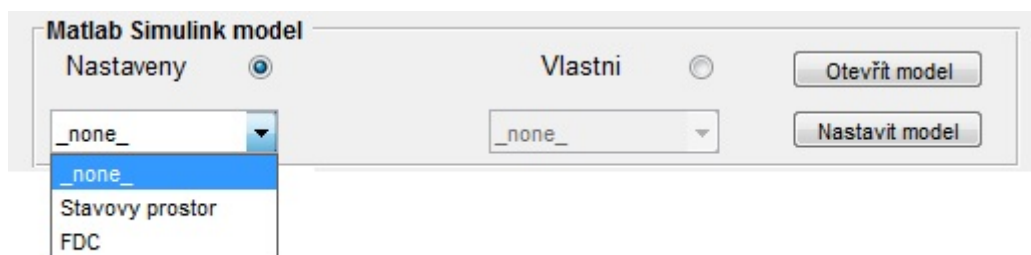
je při příštím spuštění znovu načíst.

Data se ukládají pomocí tlačítka "Uložit", po jehož spuštění se zobrazí dotazovací okno na název souboru (obrázek 4.6). Tlačítkem OK se ve složce .../VDVL/database vytvoří nová složka se zadaným názvem a do ní se uloží 7 souborů ve formátu .csv obsahující aktuální data zapsaná do tabulek. Seznam dostupných letadel v rolovacím menu ve spodní části pole "Akce" se automaticky rozšíří o tento název.

Při načítání dat z databáze je třeba vybrat z rolovacího menu daný název souboru a tlačítkem načíst uložená data.

4.3 Propojení s Matlab Simulink

Program VDVL byl navržen tak, aby bylo možné přímo z jeho uživatelského prostředí otevírat Simulink. K otevírání modelů slouží pole v základním rozhraní nazvané Matlab Simulink model (obrázek 4.7).



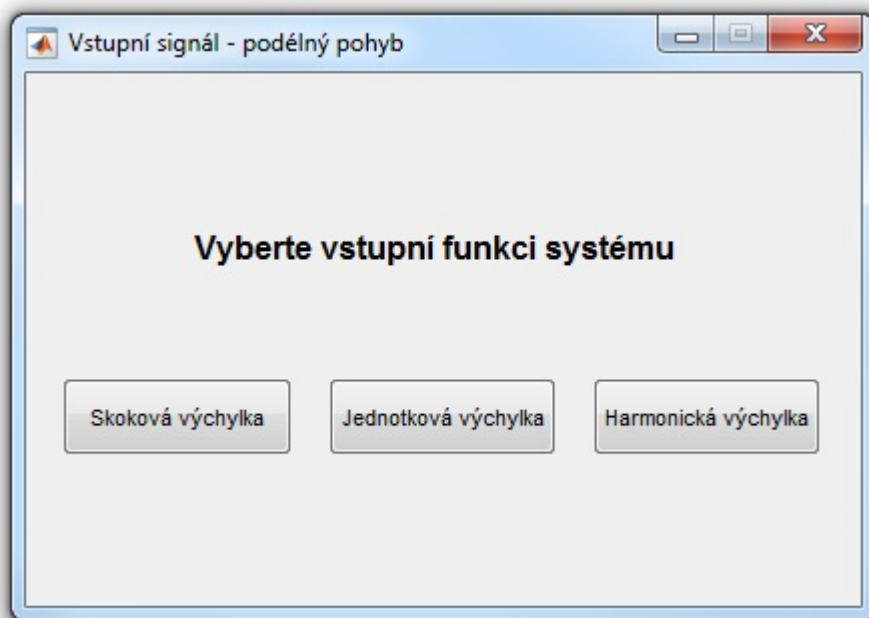
Obr. 4.7: Pole pro otevírání Simulink modelů v základním rozhraní programu VDVL

V menu "Nastaveny" lze vyvolat předdefinované modely simulující dynamiku letounu pomocí stavového prostoru, nebo model vytvořený na základě FDC (viz. kapitola 3). V menu "Vlastni" je k dispozici tutoriál, ve kterém je popsán postup návrhu modelu pro řešení odezvy vstupního signálu na základě stavového prostoru, případně lze otevřít nový prázdný Simulink dokument.

4.3.1 Předdefinované modely na základě stavového prostoru

V rolovacím menu pod výběrem "Nastaveny" je pro spuštění předdefinovaných modelů pro řešení odezvy na výchylku výškového kormidla za použití teorie stavového prostoru potřeba nastavit možnost Stavový prostor. V rámci programu jsou předdefinované tři možnosti vstupního signálu do modelu, na které se program dotazuje po spuštění zapínání prostředí Simulink tlačítkem Otevřít model. Vybrat lze

mezi vstupním signálem ve formě skokové výchylky, jednotkové výchylky a harmonické funkce, jak je zřejmé na obrázku 4.8, na kterém je dotazovací okno, které se otevře při spouštění Simulink modelu.



Obr. 4.8: Dotazovací okno na vstupní výchylku výškového kormidla předdefinovaného modelu

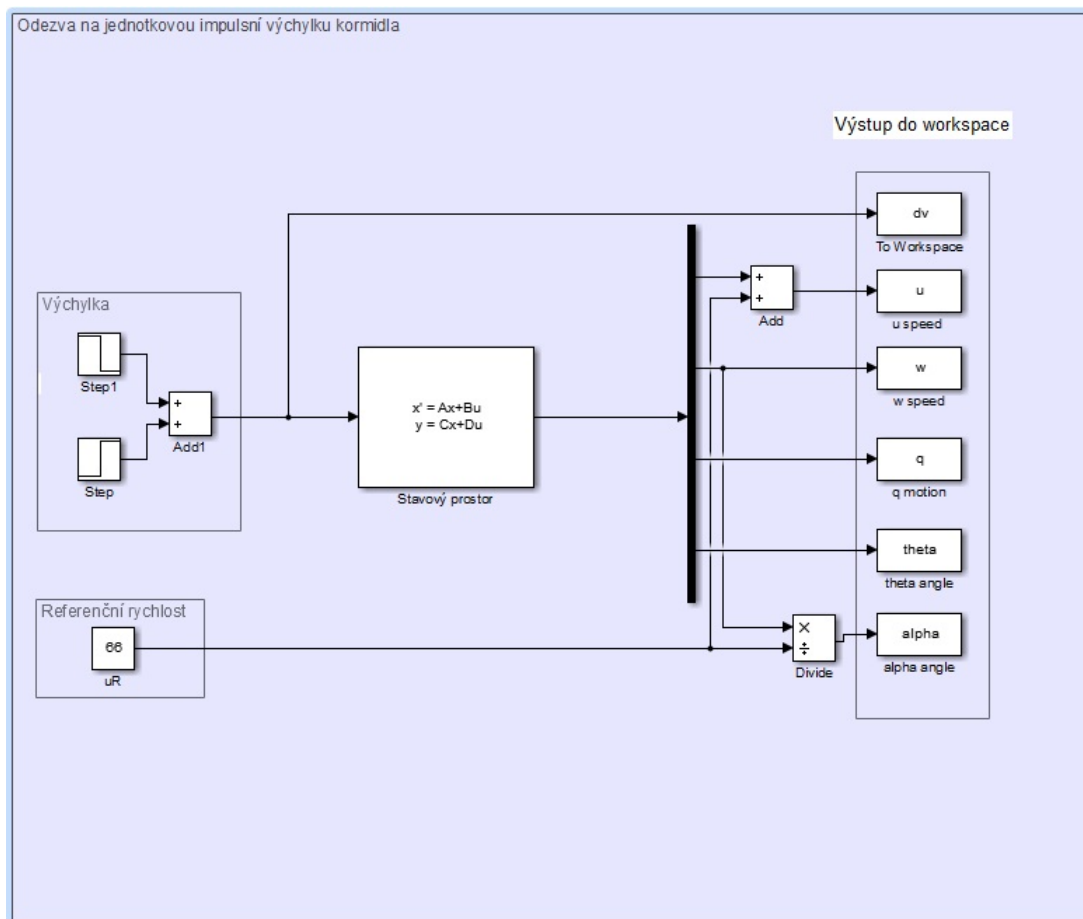
Výběrem jedné ze tří možností uživatel spustí otevření příslušného modelu. Pokud již není inicializovaný simulink, může tato akce trvat několik sekund, během kterých je v základním rozhraní v poli "*Výsledkové okno a probíhající procesy*" text, informující o otevření simulinku.

Jednotlivé modely jsou identické, liší se pouze v systému bloků, které definují vstupní výchylku kormidla. Na obrázku 4.9 je model, který se otevře při zvolení jednotkové vstupní výchylky výškového kormidla. Model se skládá z částí reprezentující vstupní výchylku, stavový prostor a výstup ze systému. Všechny výstupy jsou přenášeny do workspace pomocí bloku *To workspace* pod názvy proměnných veličin. Výstup podélné rychlosti Δu [m/s] ze stavového prostoru je zvětšen o referenční rychlost u_R [m/s], která je definována automaticky při spouštění modelu podle nastavení v tabulce *Referenční hodnoty*. Rychlost klesání w [m/s] je přepočtena na názornější úhel náběhu α vzorcem 4.1.

$$\alpha(t) = \frac{w(t)}{u_R} \quad (4.1)$$

Součástí výstupních hodnot je také výchylka výškového kormidla v čase - $dv(t)$.

Výsledky se do workspace ukládají ve formátu *timeseries*, kde každá proměnná obsahuje dva sloupce, z nichž první je vektor časů a druhý vektor daných hodnot. Každý řádek takové matice pak odpovídá jednomu kroku výpočtu.

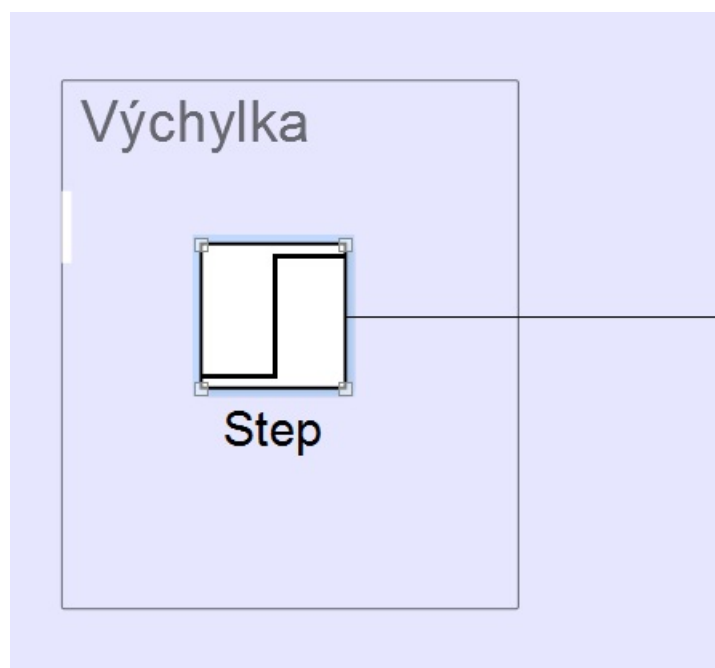


Obr. 4.9: Simulink model pro jednotkový skok výškového kormidla

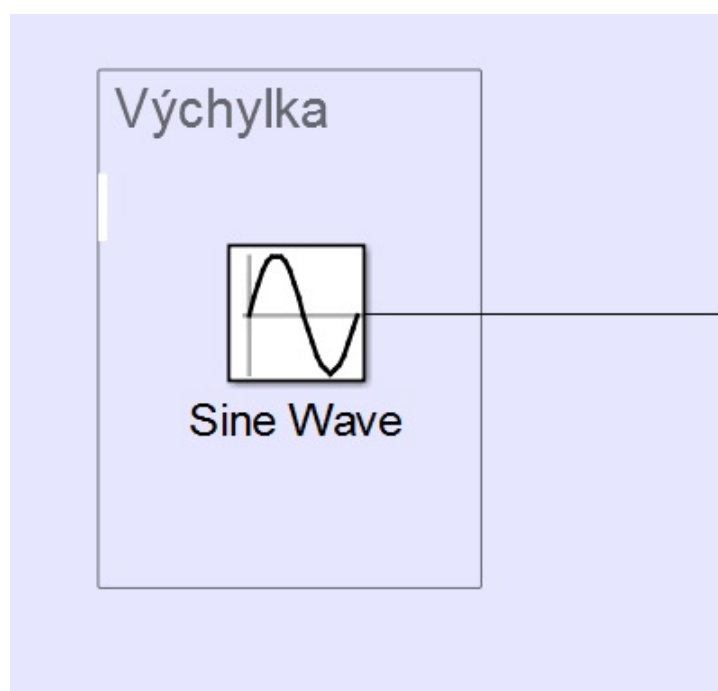
Názvy proměnných nesmí být změněny z důvodu jejich využití při vizualizaci výsledků popsané v kapitole 4.4.

Výchylka kormidla jednotkového impulsu na obrázku 4.9 je realizována pomocí dvou bloků *step*, jejichž hodnoty jsou stejné velikosti, ale opačných znamének a času změny hodnoty tak, aby se jejich hodnoty po době rovné rozdílu definovaných časů navzájem vyrušily.

V případě skokové výchylky výškového kormidla je vstup pouze jeden blok *step* (4.10) a harmonická výchylka je reprezentována blokem *sine wave* (4.11). Parametry vstupních hodnot lze libovolně měnit rozbalením vlastností bloků, jak je popsáno v kapitole 2.1.2.



Obr. 4.10: Jednotková skokotvá výchylka kormidla



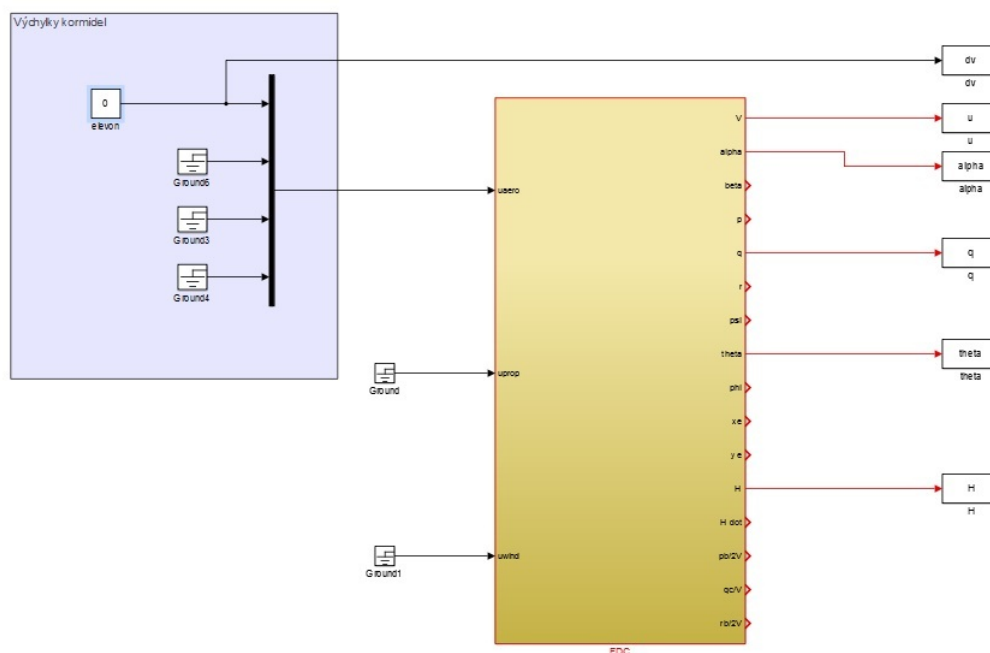
Obr. 4.11: Harmonická výchylka kormidla

4.3.2 Předdefinovaný model na základě FDC

Model na základě FDC vyvolatelný programem VDVl je na obrázku 4.12. Stejně jako modely ve stavové prostoru využívá tento model charakteristiky definované v maticích a vektorech ve *workspace*. Jedná se o matice AM, EM, GM1, GM2 a vektory Xinco a xfix, které lze nastavit pomocí funkce [AM, EM, GM1, GM2, xinco, xfix] = Příklad1_FDC, napsané do Command Window.

Matice AM obsahuje bezrozměrné derivace letounu, EM řiditelnostní derivace, GM1 je matice geometrických a hmotnostních parametrů, GM2 matice setrvačných sil. Xinco je vektor počátečních podmínek a xfix je vektor, který definuje, které hodnoty jsou variabilní a které zůstávají během simulace konstantní. Parametry jsou defaultně nastaveny pro letoun definovaný v databázi jako Příklad1 a pomocí vektoru xfix je výstup degradován na podélný pohyb letounu.

Změnu parametrů lze provést otevřením skriptu `Priklad1_FDC` v cílové složce `...VDVL/tmp/Priklad1_FDC.m`. Bezrozměrné derivace a geometrické charakteristiky jsou na řádcích 7 – 49, vektory `xfix` a `xinco` na řádcích 137 a 142.

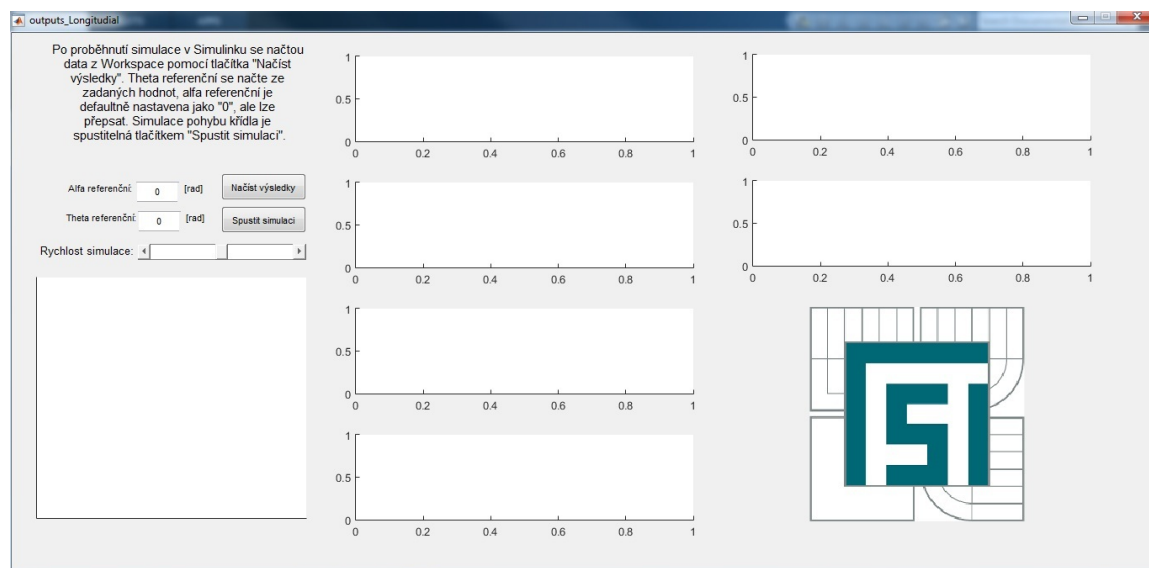


Obr. 4.12: Předdefinovaný model FDC

Výstupní veličiny z tohoto modelu jsou opět předávány do workspace pomocí bloků *To workspace*.

4.4 Vizualizace výsledků

Výsledkové okno programu lze otevřít v poli Výsledky pomocí tlačítka "Výsledkové okno". Na obrázku 4.13 je okno, které se po aktivaci tlačítka otevře. V levé části se nachází ovládací panel, pomocí kterého lze nastavit referenční úhly letounu, načítat výstupní data ze Simulink modelů uložené ve *workspace* a spouštět grafickou simulaci pohybu letounu.

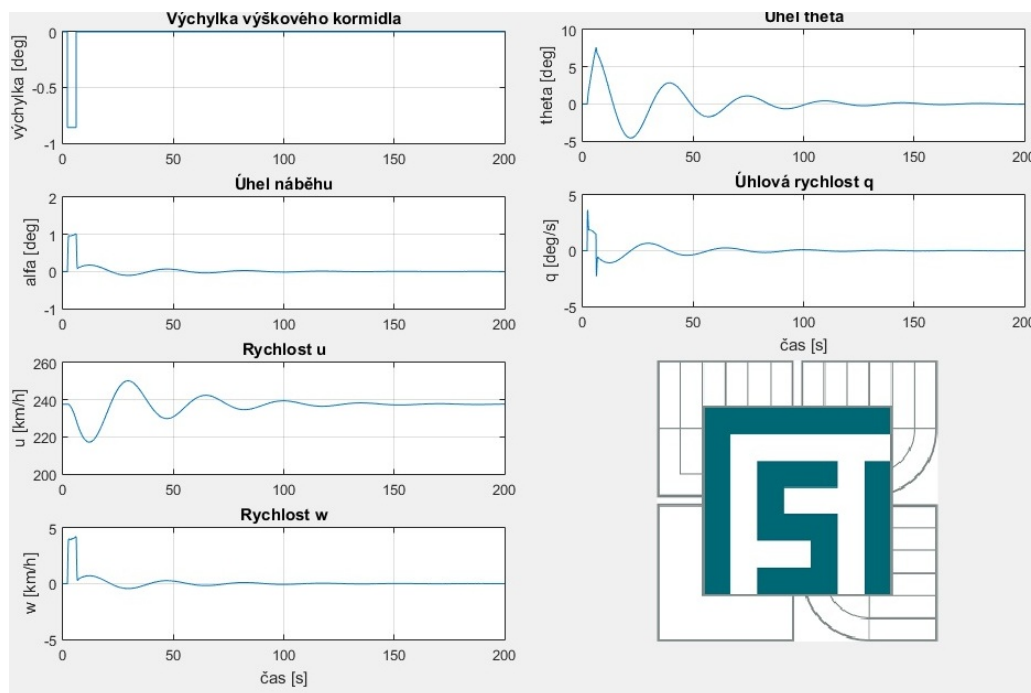


Obr. 4.13: Prázdné okno výstupů

Pokud byl spuštěn model v simulinku a ve *workspace* jsou definované výstupy z modelu, je možné načíst tyto hodnoty do grafů ve výstupním okně tlačítkem "Načíst data". Automaticky se vytvoří popisy a rozsahy os v grafech a zobrazí se v nich průběhy hodnot v závislosti na čase simulace. V případě že byl ze základního rozhraní jako poslední spuštěn model FDC, pak nejnižší umístěný graf ukazuje závislost výšky H [km] na čase. Výpočtem pomocí stavového prostoru žádná informace o výšce nepřichází (musela by být dopočítávána z rychlosti klesání letounu a ve zmíněném grafu se zobrazuje rychlost letounu v ose Z letadlové souřadné soustavy w [m/s]).

Na obrázku 4.14 je detail na grafy výsledkového okna s načtenými výstupy ze simulace ve stavovém prostoru pro jednotkovou impulzní výchylku výškového kormidla. Jednotky pro zobrazení byly přepočítány na běžnější km/h z m/s, stupně z radiánů a stupně/s z rad/s. V grafech je zobrazena výchylka kormidla v čase, úhel náběhu α , rychlost v ose x letadlové souřadné soustavy u [m/s], rychlost v ose z letadlové souřadné soustavy w [m/s], úhel letounu proti horizontu θ a úhlová rychlost kolem osy Y letadlové souřadné soustavy q [°/s].

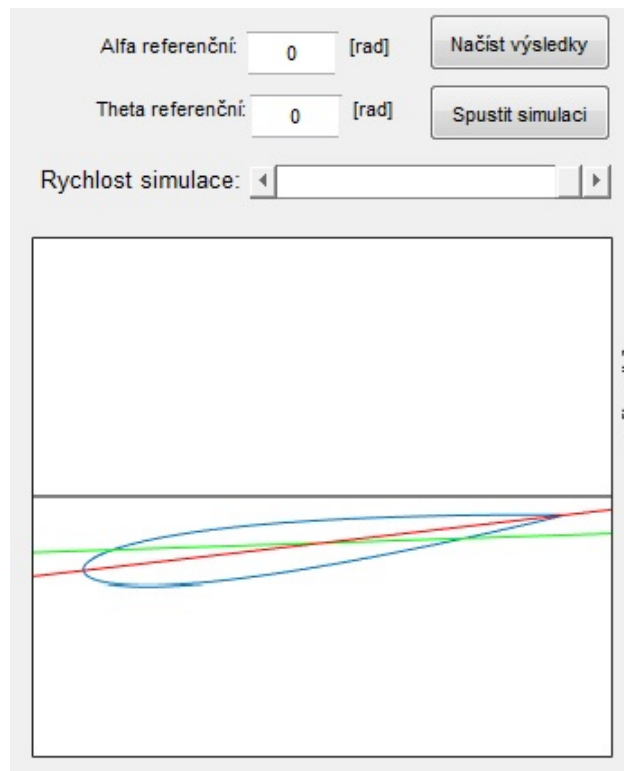
Načtené hodnoty lze při opětovném spuštění simulace v Simulink modelu a přepsáním výstupních hodnot ve *workspace* jednoduše nahradit novým načtením dat tlačítkem "Načíst data".



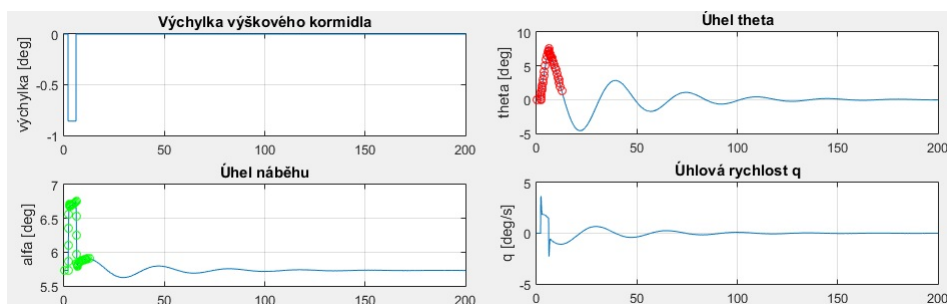
Obr. 4.14: Okno s importovanými výsledky simulace

Na obrázku 4.15 je detail levé části okna, se spuštěnou grafickou simulací pohybu křídla letounu. Tu lze spustit po načtení dat tlačítkem "Spustit simulaci" a její rychlost regulovat posuvem posuvníku. Černá vodorovná čára značí horizont, zelená udává směr pohybu nabíhajícího proudu a červená čára představuje prodlouženou tětivu profilu. Zobrazený profil se otáčí oproti horizontu o úhel θ , zatímco zelená čára rotuje kolem profilu podle okamžitého úhlu náběhu α . Podle rychlosti klesání se při simulaci vypočítává změna výšky a profil mění svoji vertikální polohu v okně, podle poměru změny výšky k rozsahu osy Y. V grafech pro úhel náběhu a úhel θ se během grafické simulace přikreslují barevné značky, aby bylo zřejmé v jakém časovém okamžiku se simulace právě nachází (obrázek 4.16).

V případě, že byla grafická simulace spuštěna pro výstup z modelu FDC, není třeba nastavovat referenční úhly, jelikož tento model nepracuje pouze s odchylkami, jako stavový prostor a do výstupových hodnot ukládá přímo velikosti jednotlivých veličin.



Obr. 4.15: Grafická simulace pohybu letounu



Obr. 4.16: Přikreslování značek do grafu při průběhu grafické simulace

5 IMPLEMENTACE PROGRAMU VDL

Program VDL tvoří téměř 50 funkcí a skriptů napsaných ve skriptovacím programovacím jazyku Matlab, které jsou rozděleny do jednotlivých skupin podle svojí primární funkce. Celý program je uložen ve složce VDL, která obsahuje podsložky pro jednotlivé skupiny souborů. Jejich výčet a stručná charakteristika v tabulce 5.1.

Tab. 5.1: Rozdělení souborů tvořících program VDL do složek podle jejich funkce

složka	obsah
data	struktury některých proměnných a datové soubory
database	databáze uložených hodnot letounů
fig	obrázky použité v programu
grafic	funkce pro spouštění grafické simulace
lib	knihovna funkcí definujících grafické rozhraní a výpočtové třídy
sim	knihovna modelů vytvořených v Simulinku
tmp	některé pomocné funkce a nápověda

Složky, které nejsou v tabulce výše byly vytvořeny Matlabem při iniciaci některých částí programu.

Funkce, které nejsou zařazeny do žádné z podsložky jsou iniciační a slouží k předdefinování globálních proměnných, nastavení vyhledávacích cest a vytvoření datových struktur pro program.

5.1 Inicializace programu

VDL se spouští přes příkazový řádek v prostředí Matlab příkazem *VDL*, nebo *VDLinit*. Po zadání příkazu program nejprve zjistí umístění složky VDL a přidá podsložky z tabulky 5.1 a jejich obsahy do seznamu cest pro vyhledávání volaných funkcí. Dále definuje datové objekty, které reprezentují obsahy zadávacích tabulek programu a vytvoří globální strukturu *Dat*, v níž uchovává zadané hodnoty v tabulkách parametrů, referenčních hodnot, derivací a matic stavového prostoru. Dále definuje globální proměnnou *Open*, která napomáhá identifikovat, jestli je daná tabulka otevřená (díky tomu je možné vyhnout se vícenásobnému otevření některého z oken programu).

Po vykonání zmíněných inicializačních operací zavolá funkci *open_menu*, která vytvoří základní uživatelské rozhraní programu.

Při spouštění se právě provádějící akce pro informaci vypisují do příkazového řádku Matlabu.

5.2 Uživatelské prostředí

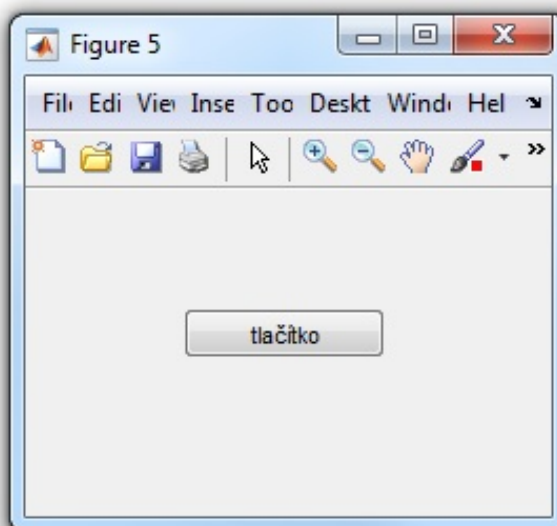
Okna uživatelského prostředí programu jsou kromě výsledkového okna naprogramovány ručně, kdy jsou jednotlivé objekty v okně a jejich vlastnosti (jako poloha, velikost, volané funkce a podobně) definované přímo v kódu bez použití Matlab guide (2.1.3).

V prvním kroku při vytváření nového okna, se vyvolá prázdný objekt *figure* a do něho se postupně přidávají další grafické objekty s definovanými vlastnostmi. Aby bylo načítání rychlejší, není toto nové okno viditelné až do doby, kdy jsou všechny objekty vytvořené (vlastnost "Visible" je nastavená na "off").

Přidání dalšího objektu je provedeno vyvoláním jedné z grafických funkcí (*uicontrol*, *uibuttongroup*...) v jejichž parametrech se zapíše vlastnosti objektu ve formátu

funkce(parent, "par1", "nast1", "par2", "nast2", ... "parn", "nastn").

Parent vyjadřuje objekt, ke kterému je nový grafický objekt přiřazen, par jsou parametry a nast je nastavení daného parametru. Mezi nejdůležitější používané parametry patří "Style", který určuje jaký objekt bude vytvořen (pushbutton - tlačítko, popupmenu - rozbalovací menu...), "Position", který určuje polohu a velikost příslušného objektu, "Tag", podle kterého se pomocí funkce *findobj('Tag', 'name')* vyhledávají vlastnosti objektu v dalších částech programu a "Callback", určující funkci, kterou daný objekt (například tlačítko) po akci vyvolává.



Obr. 5.1: Příklad vytvoření objektu pushbutton funkcí *uicontrol*

Na obrázku 5.1 je příklad vytvoření tlačítka funkcí `uicontrol`. Na obrázku 5.2 je skript, který tlačítko vytváří. Je zde patrný způsob zadávání vlastností a iniciace objektu.

```
n = figure(5);

push_zadzob = uicontrol(n, 'Style', 'pushbutton',...
                        'Units', 'centimeters',...
                        'String', 'tlačítko',...
                        'Position', [2 2 2.5 0.6]);
```

Obr. 5.2: Skript vytvářející tlačítko na obrázku 5.1

Rozsáhlejší informace o tvorbě grafických prostředí v Matlabu lze najít v literatuře [10].

Grafické prostředí výsledkového okna bylo vytvořené pomocí Matlab guide, vzhledem k jednodušší práci umisťováním a nastavováním více grafů do jednoho okna.

Součástí programu VDVL je množství oken, které nemusejí být permanentně otevřené. Aby nedocházelo k vícenásobnému vytvoření stejného okna při jeho opakovaném vyvolávání příslušným tlačítkem, byla při iniciaci programu definována globální proměnná *Open*. Je to vektor nul, ve kterém je každému pořadí přiřazeno jedno okno programu. Při otevření příslušného okna se nula na daném místě ve vektoru přepíše na hodnotu 1. Před každou inicializací nového okna je tak možné kontrolovat, jestli už není otevřené a v případě že ano, přesune se pouze do popředí na obrazovce. Při zavírání okna je přiřazená hodnota ve vektoru *Open* opět nastavena na nulu.

5.3 Výpočty parametrů letounu

Rovnice, které jsou potřebné pro výpočet bezrozměrný derivací, rozměrových derivací a matic stavového prostoru jsou definovány jako metody tříd, uložených ve složce */VDVL/lib/calculate*. Při výpočtu jsou do těchto funkcí zasílány aktuální data zapsané v tabulkách (respektive v globální struktuře *Dat*) pomocí funkcí *write_tabulka* (uložené ve složce */VDVL/lib/UIs*), které transformují datové typy hodnot do formátů použitelných při matematických operacích. Hodnoty vypočítané třídami jsou automaticky uloženy do globální struktury *Dat* a tabulky v zadávacím okně jsou po proběhnutí výpočtu automaticky aktualizovány. V případě, že je výpočet prováděn z

jiného okna než ze zadávacího a současně je otevřené okno s počítanými hodnotami, pak je nutné okno aktualizovat ručně tlačítkem "aktualizovat".

Výpočetní třídy programu VDVL jsou:

- *dimensionless_derivative* - obsahuje vztahy pro výpočet bezrozměrných derivací
- *dimensional_derivatives* - obsahuje vztahy pro výpočet rozměrových derivací
- *coefficients_input_matrix_Longitudinal* - obsahuje vztahy pro výpočet matice vstupů **B** stavového prostoru
- *coefficients_system_matrix_Longitudinal* - obsahuje vztahy pro výpočet matice systému **A** stavového prostoru

Výpočetní vztahy ve zmíněných třídách jsou pro přehled zapsány do tabulek 5.2, 5.3 a 5.4. Třídy pro výpočet matic stavového prostoru využívají vzorců definovaných v tabulce 1.1 v kapitole 1.1.2.

Tab. 5.2: Výpočet bezrozměrných derivací ve třídě *dimensionless_derivative*

[1]

bezrozměrná derivace	vztah pro výpočet
C_{X_u}	$C_{F_u} \cdot (1 - C_{D_{CFR}})$
C_{Z_u}	$C_{F_u} \cdot C_{L_{CFR}}$
C_{m_u}	$C_{F_u} \cdot C_{m_{CFR}}$
C_{X_α}	$C_{L_R} \cdot C_{D_\alpha}$
C_{Z_α}	$- C_{L_R} \cdot C_{D_R}$
C_{m_α}	$C_{L_\alpha} \cdot (\bar{x}_T - \bar{x}_A)$
C_{X_q}	0
C_{Z_q}	$-2 \cdot C_{L_{VOP\alpha}} \cdot k_{VOP} \cdot \tilde{V}_{VOP}$
C_{m_q}	$1.1 \cdot C_{Z_q} \cdot \frac{\tilde{l}_{VOP}}{c_{AK}}$
$C_{X\dot{\alpha}}$	0
$C_{Z\dot{\alpha}}$	$C_{Z_q} \cdot \frac{\partial \epsilon}{\partial \alpha}$
$C_{m\dot{\alpha}}$	$C_{Z\dot{\alpha}} \cdot \frac{\tilde{l}_{VOP}}{c_{AK}}$
$C_{Z_{\delta V}}$	$- C_{L_{VOP\alpha}} \cdot k_{VOP} \cdot \frac{S_{VOP}}{S_K} \cdot \frac{\partial \alpha}{\partial \delta_V}$
$C_{m_{\delta V}}$	$- C_{L_{VOP\alpha}} \cdot k_{VOP} \cdot \tilde{V}_{VOP} \cdot \frac{\partial \alpha}{\partial \delta_V}$
$C_{m_{\dot{\delta V}}}$	0

Ve vztazích v tabulce 5.2 se vyskytují některé hodnoty, které si třída dopočítává při volání konstruktoru. Výpočty provádí podle vzorců v tabulce 5.3.

Tab. 5.3: Funkce pro výpočet bezrozměrných derivací ve třídě *dimensionless_derivative*

[4]

hodnota	vztah pro výpočet
C_G	$\frac{2 \cdot m \cdot n \cdot g}{\rho_R \cdot V_R \cdot S_K}$
C_{LR}	$\frac{2 \cdot m \cdot g \cdot n}{\rho_R \cdot V_R^2 \cdot S_K}$
C_{DR}	$\frac{C_{D0} + C_{LR}^2}{A_K \cdot e_0 \cdot \pi}$
C_{FR}	$C_{DR} + C_G \cdot \sin(\gamma)$
C_{Fu}	$-3 \cdot C_{FR}$
$C_{L\alpha}$	$C_{LK\alpha} + C_{L_{VOP\alpha}} \cdot k_{VOP} \cdot \frac{S_{VOP}}{S_K} \cdot \frac{\partial \epsilon}{\partial \alpha}$
\tilde{l}_{VOP}	$l_{VOP} + (0.25 \cdot c_{AK}) - (\bar{x}_T \cdot c_{AK})$
\bar{l}_{VOP}	$l_{VOP} + (0.25 \cdot c_{AK}) - (\bar{x}_A \cdot c_{AK})$
$C_{LR\alpha}$	$C_{L\alpha} \cdot \frac{l_{VOP}}{\bar{l}_{VOP}}$
$C_{DR\alpha}$	$\frac{2 \cdot C_{LR} \cdot C_{LR\alpha}}{\pi \cdot A_K \cdot e_0}$
\tilde{V}_{VOP}	$\frac{S_{VOP} \cdot l_{VOP}}{S_K \cdot c_{AK}}$

Tab. 5.4: Výpočet rozměrových derivací ve třídě *dimensional_derivative*

[1]

rozměrové derivace	vztah pro výpočet
X_u	$\frac{1}{2} \cdot \rho_R \cdot V_R \cdot S_K + C_G \cdot \rho_R \cdot V_R \cdot S_K \cdot \sin(\theta_R)$
X_w	$\frac{1}{2} \cdot \rho_R \cdot V_R \cdot S_K \cdot C_{X\alpha}$
X_q	$\frac{1}{4} \cdot C_{Xq} \cdot \rho_R \cdot V_R \cdot S_K \cdot c_{AK}$
$X_{\dot{w}}$	$\frac{1}{4} \cdot C_{X\dot{\alpha}} \cdot \rho_R \cdot S_K \cdot c_{AK}$
Z_u	$\rho_R \cdot V_R \cdot S_K \cdot (\frac{1}{2} \cdot C_{Zu} - C_G \cdot \cos(\theta_R))$
Z_w	$\frac{1}{2} \cdot \rho_R \cdot V_R \cdot S_K \cdot C_{Z\alpha}$
Z_q	$\frac{1}{4} \cdot C_{Zq} \cdot \rho_R \cdot V_R \cdot S_K \cdot c_{AK}$
$Z_{\dot{w}}$	$\frac{1}{4} \cdot C_{Z\dot{\alpha}} \cdot \rho_R \cdot S_K \cdot c_{AK}$
M_u	$\frac{1}{2} \cdot \rho_R \cdot V_R \cdot S_K \cdot C_{m_u} \cdot c_{AK}$
M_w	$\frac{1}{2} \cdot \rho_R \cdot V_R \cdot S_K \cdot C_{m_\alpha} \cdot c_{AK}$
M_q	$\frac{1}{4} \cdot C_{m_q} \cdot \rho_R \cdot V_R \cdot S_K \cdot c_{AK}^2$
$M_{\dot{w}}$	$\frac{1}{4} \cdot C_{m_{\dot{\alpha}}} \cdot \rho_R \cdot V_R \cdot S_K \cdot c_{AK}^2$
Z_{δ_V}	$\frac{1}{2} \cdot C_{Z_{\delta V}} \cdot \rho_R \cdot V_R^2 \cdot S_K \cdot c_{AK}$
M_{δ_V}	$\frac{1}{2} \cdot C_{m_{\delta V}} \cdot \rho_R \cdot V_R^2 \cdot S_K \cdot c_{AK}$
$M_{\dot{\delta}_V}$	$\frac{1}{4} \cdot C_{m_{\dot{\delta}_V}} \cdot \rho_R \cdot V_R^2 \cdot S_K \cdot c_{AK}$

6 VYUŽITÍ VDWL VE VÝUCE PŘEDMĚTU MECHANIKA LETU II

Program VDWL byl vytvořen pro účely snadnějšího pochopení matematického popisu dynamických vlastností letounu pomocí bezrozměrných a rozměrových derivací, využívání stavového prostoru k zjišťování odezev na výchylku kormidel a pochopení základů blokového programování v prostředí Matlab Simulink. Jeho uplatnění při výuce předmětu Mechanika letu II může spočívat v pozorování změn v dynamice letounu při změně některých parametrů, případně derivací letounu. Díky uživatelskému prostředí, které tento program nabízí, nemusí mít uživatel hlubší znalosti programování v jazyku Matlab a přesto využívat část jeho výpočetního potenciálu ke zkoumání dynamiky letounu. Grafický výstup programu seřazuje výstupy dynamických simulací do jednoho okna, což je pro studenta osvojujícího si základy zmiňovaného předmětu přehlednější a s grafickou simulací pohybu křídla mu může program poskytnout lepší představu o tom co znamená stabilita letounu.

6.1 Příklady do výuky

V rámci této práce byly vytvořeny návrhy příkladů do cvičení z předmětu Mechanika letu II, využívající program VDWL. Příklady jsou zaměřeny na práci v prostředí Matlab Simulink.

V prvním příkladu (kapitola 6.1.1) se student naučí základům blokového programování a zpracování výstupů ze simulace. Zároveň si osvojí práci s programem VDWL.

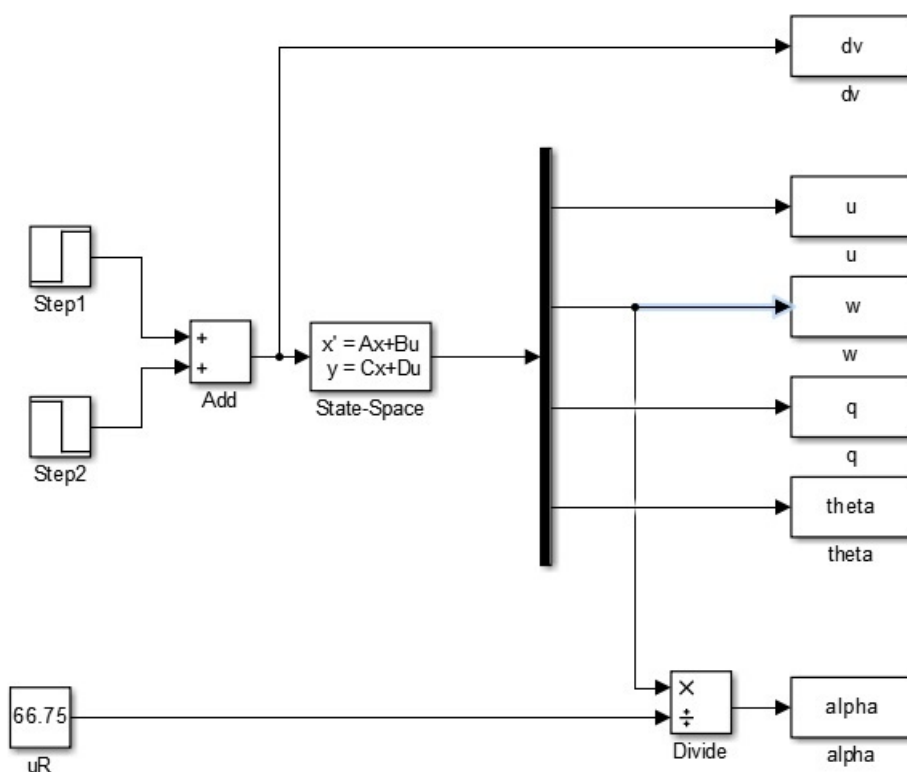
Druhý navrhovaný příklad (kapitola kapitola 6.1.2) využívá dynamického modelu FDC, zakomponovaného v programu. Je navržen k pochopení rozdílu, mezi chováním letounu v modelu stavového prostoru, kdy je pohyb letounu uvažován jako rovnovážný až do poruchy a letovým vlastnostem letounu v reálných podmínkách, kterým se svojí podstatou model FDC oproti stavovému prostoru více blíží.

6.1.1 Model stavového prostoru

Součástí programu VDWL je tutoriál, který lze otevřít po jeho nastavení v rozbalovací nabídce v sekci "Matlab Simulink model - vlastní", nebo tlačítkem "Tutorial" ve spodní části základního rozhraní. Jeho spuštěním se otevře prázdné okno pracovního prostředí Simulinku a soubor Tutorial.pdf. Tutorial tvoří návod na vytvoření základního modelu pro řešení podélné stability pomocí stavového prostoru, jeho

správné nastavení pro výstupy z VDVL, spuštění simulace a zobrazení výsledků ve výsledkovém okně.

Návod na sestavení modelu je dostatečně podrobný, aby podle něho bylo možné model sestavit bez předešlých zkušeností s blokovým programováním v prostředí Matlab Simulink. Blok po bloku popisuje sestavení modelu, včetně základů nastavení parametrů jednotlivých bloků. Popisuje základní funkci jednotlivých použitých bloků a jejich význam v konečném modelu.



Obr. 6.1: Model v prostředí Matlab Simulink vytvořený pomocí tutoriálu

Student vytvoří model se stavovým prostorem v simulinku podle návodu v tutoriálu programu (obrázek 6.1) a pro zadaný letoun a parametry vytvoří simulaci odezvy na poruchu v podobě výchylky výškového kormidla. Z výsledků simulace letounu pak určí součinitel doznívání a kruhovou frekvenci kmitání. Dále vypočítá potřebnou dobu k utlumení na 1/2 amplitudy a počet kmitů do utlumení.

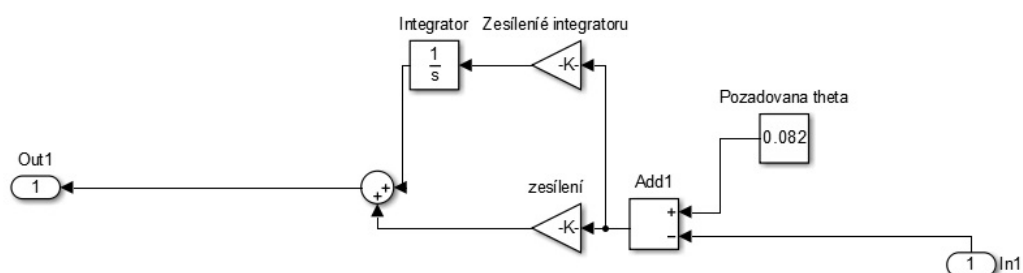
Výstupem tohoto cvičení může být protokol obsahující grafy jednotlivých výstupních veličin a porovnání výsledků součinitele doznívání, kruhové frekvence, potřebného času a počtů kmitů do utlumení získaných výpočtovou metodou z charakteristické rovnice pro rychlý podélný kmitavý pohyb s grafickým odhadem ze získaných grafů.

6.1.2 Regulátor úhlu θ

V rámci tohoto příkladu student využívá přednastavený model FDC, vyvolatelný z prostředí VDVL. Pro letoun zadaný v příkladu v kapitole 6.1.1 vytvoří PI, případně PID regulátor úhlu θ . Nastavení modelu FDC je prováděno skriptem `[AM, EM, GM1, GM2, xinco, xfix] = Prik1_FDC`, podle návodu programu vyvolatelného tlačítkem "Návod" v dolní části základního rozhraní, do kterého zapíše parametry zadaného letounu.

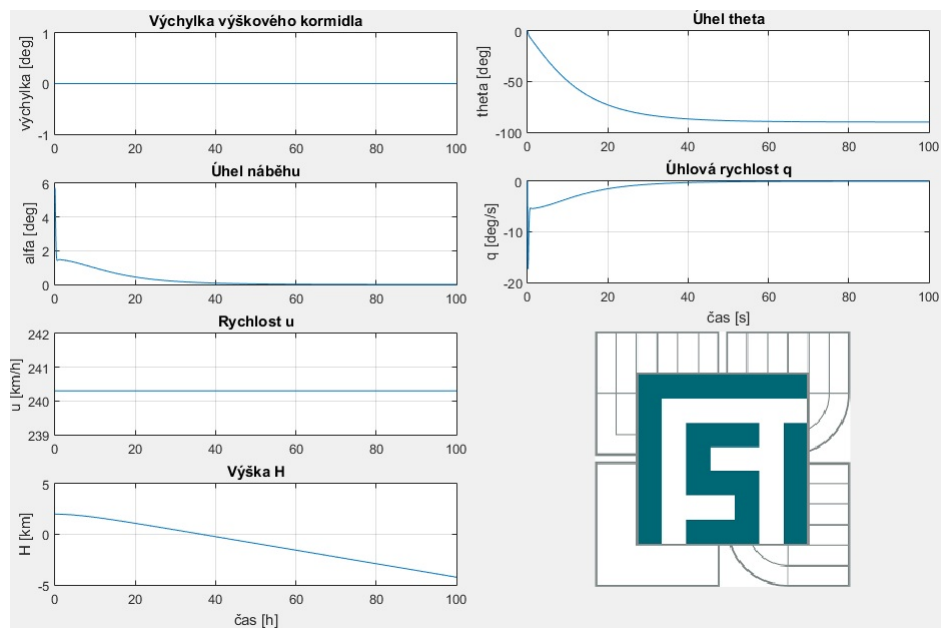
Rychlost regulovaného letounu bude pro jednoduchost uvažována konstantní. To je docíleno fixací této hodnoty pomocí vektoru `xfix`, který je ve zmíněném skriptu předdefinovaný. Vzhledem ke skutečnosti, že regulace ostatních výstupních hodnot vyžaduje více zapojených regulátorů, byl za regulovanou hodnotu zvolen úhel θ .

Na obrázku 6.2 je příklad PI regulátoru pro regulaci úhlu θ . Regulátor byl sestaven pro letoun Prik1 z databáze programu VDVL a je uložený v modelu FDC jako referenční. Požadovaný úhel θ je zvolený tak, aby letoun udržoval co nejvíce horizontální let. Výstupy pro neregulovaný a regulovaný letoun jsou na obrázcích 6.3 a 6.4. Na obrázku 6.5 je zapojení regulátoru do systému.

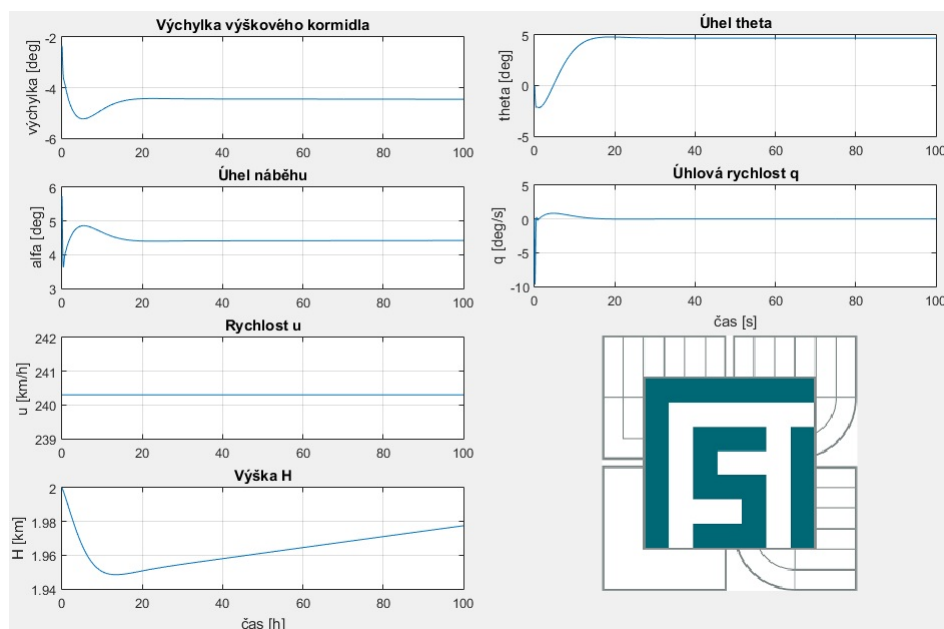


Obr. 6.2: Příklad PI regulátoru pro regulaci úhlu θ

Pomocí tohoto příkladu by měl student pochopit, jak se chová neřízený letoun mimo rovnovážné režimy letu a funkci regulátoru jakožto úvod do problematiky automatizace řízení letounu. Výstupem ze cvičení může být protokol obsahující postup návrhu regulátoru a zjištěné ideálního zesílení jednotlivých prvků.



Obr. 6.3: Výstup pro neřízený letoun bez regulace úhlu θ při konstantní nastavení nulové výchylky výškového kormidla



Obr. 6.4: Výstup pro letoun s regulací úhlu θ



7 ZÁVĚR

V teoretické části diplomové práce byl popsán způsob matematického popisu letové dynamiky letounu. Byly popsány souřadnicové soustavy používané pro tento popis, naznačeno odvození pohybových rovnic a matic stavového prostoru. Dále byl stručně popsán program Matlab, jeho část Simulink a aplikace modelu pro řešení letových vlastností FDC.

V praktické části je popsán pro účely této práce vytvořený program VDVL pro výpočet stabilitních a říditelnostních derivací a matic stavového prostoru pro podélný pohyb letounu. V závěru diplomové práce byly navrženy příklady do cvičení z předmětu Mechanika letu II zaměřené na práci s programem VDVL a prostředím Matlab Simulink.

LITERATURA

- [1] DANĚK, Vladimír. *Mechanika letu II*. 1. vyd. Brno: Akademické nakladatelství CERM, 2011, 334 s. ISBN 8978-80-7204-761-1.
- [2] DANĚK, Vladimír. *Mechanika letu I: Letové výkony*. první. Brno: Akademické nakladatelství CERM, 2009, 293 s. ISBN 978-80-7204-659-1.
- [3] BROŽ, Václav, Vladimír DANĚK a Karol FILAKOVSKÝ. *Základy aerodynamiky: studijní modul 8*. Brno: Akademické nakladatelství CERM, 2003, 163 s. Učební texty dle předpisu JAR-66. ISBN 80-720-4316-1.
- [4] MATĚJŮ, Jiří. *Hodnocení stability a říditelnosti čtyřmístného, jednomotorového letounu*. Brno, 2013. 66 s. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=67636. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství. Vedoucí práce doc. Ing. Vladimír Daněk, CSc.
- [5] KARBAN, Pavel. *Výpočty a simulace v programech Matlab a Simulink*. 1. vyd. Brno: Computer Press, a.s., 2006, 220 s. ISBN 80-251-1301-9.
- [6] HORA, Petr a Blanka HERINGOVÁ. *MATLAB: Díl 1: Práce s programem* [online]. Plzeň: H-S, 1995 [cit. 2016-02-15]. Dostupné z: <http://www.cdm.cas.cz/czech/hora/vyuka/mvs/tutorial.pdf>
- [7] ŠTECHA, Jan a Vladimír HAVLENA. *Teorie dynamických systémů*. Vyd. 2. Praha: České vysoké učení technické, 1999, iv, 248 s. ISBN 80-010-1971-3.
- [8] MATLAB Documentation. [online]. [cit. 2016-02-15]. Dostupné z: <http://www.mathworks.com/help/matlab/>
- [9] RAUW, Marc. *FDC 1.2: A Simulink Toolbox for Flight Dynamics and Control Analysis*. 2. vyd. 2001. ISBN 90-807177-1-1. Dostupné z: <http://www.dutchroll.com>
- [10] THE MATHWORKS, Inc. *Matlab: Creating Graphical User Interfaces* [online]. Revised for MATLAB 8.6 (Release 2015b). Natick, 2015 [cit. 2016-02-18]. Dostupné z: <https://www.mathworks.com/help/pdfdoc/matlab/buildgui.pdf>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

Ψ, Θ, Φ	Eulerovy polohové úhly
χ_a, γ_a, μ_a	Aerodynamické polohové úhly
x_g, y_g	Souřadnice v horizontální rovině
H	Hybnost
b	Točivost
F_a	Výsledná zrychlující síla
Q	Výsledný zrychlující moment
u, v, w	Složky výsledné rychlosti v letadlové souřadné soustavě
$\dot{u}, \dot{v}, \dot{w}$	Složky výsledného zrychlení v letadlové souřadné soustavě
q, p, r	Složky výsledné úhlové rychlosti v letadlové souřadné soustavě
$\dot{q}, \dot{p}, \dot{r}$	Složky výsledného úhlového zrychlení v letadlové souřadné soustavě
I_x, I_y, I_z	Momenty setrvačnosti k osám letounu
$I_{xy}, I_{yx}, I_{zy}, I_{xz}, I_{yz}, I_{zx}$	Deviační momenty
I	Inerční matice
g	Gravitační zrychlení
m	Hmotnost
u_R	Referenční rychlost
A	Matice soustavy stavového prostoru
B	Matice řízení stavového prostoru
α	Úhel náběhu
β	Úhel vybočení
H	Výška
x	Stavový vektor (FDC)
xinco	Vektor počátečních podmínek (FDC)

\mathbf{x}_{fix}	Vektor blokace změn parametrů (FDC)
q_{dyn}	Dynamický tlak
S	Plocha křídla
ρ	Hustota vzduchu

SEZNAM OBRÁZKŮ

1.1	Rozdělení mechaniky letu atmosférického letounu	9
1.2	Aerodynamická souřadnicová soustava (index a) a letadlová souřadnicová soustava (bez indexu) [2]	10
1.3	Eulerovy polohové úhly [2]	11
1.4	Systém řízení a) a ovládání b)	15
2.1	Uživatelské rozhraní Matlabu R2015a	17
2.2	Skupiny bloků v základní knihovně Simulinku	17
2.3	Nejčastěji používané bloky	18
2.4	Blokový model v Simulinku	19
2.5	Výstup z blokového modelu na obr. 2.4	19
2.6	Prostředí pro návrh UI [10]	20
3.1	Letoun Beaver, pro který byl původně navržen dynamický model [9] .	21
3.2	Blokový diagram struktury FDC [9]	23
3.3	Vektor polynomických funkcí pro výpočet sil a momentů [9]	24
3.4	Vektor polynomických funkcí pro výpočet sil a momentů od propulze [9]	25
3.5	Implementace dynamického modelu FDC. Upraveno [9]	26
3.6	Vnitřní struktura bloku Gravity z diagramu 3.5	27
4.1	Základní uživatelské rozhraní programu	28
4.2	Tabulky bezrozměrných derivací, rozměrových derivací a matic stavového prostoru	29
4.3	Zadávací okno programu	30
4.4	Pole akce v zadávacím okně	31
4.5	Postup při výpočtu v zadávacím okně	32
4.6	Dotazovací okno při ukládání zadaných hodnot do databáze	32
4.7	Pole pro otevírání Simulink modelů v základním rozhraní programu VDVL	33
4.8	Dotazovací okno na vstupní výchylku výškového kormidla předdefinovaného modelu	34
4.9	Simulink model pro jednotkový skok výškového kormidla	35
4.10	Jednotková skokotvá výchylka kormidla	36
4.11	Harmonická výchylka kormidla	36
4.12	Předdefinovaný model FDC	37
4.13	Prázdné okno výstupů	38
4.14	Okno s importovanými výsledky simulace	39
4.15	Grafická simulace pohybu letounu	40
4.16	Přikreslování značek do grafu při průběhu grafické simulace	40

5.1	Příklad vytvoření objektu pushbutton funkcí uicontrol	42
5.2	Skript vytvářející tlačítko na obrázku 5.1	43
6.1	Model v prostředí Matlab Simulink vytvořený pomocí tutoriálu	47
6.2	Příklad PI regulátoru pro regulaci úhlu θ	48
6.3	Výstup pro neřízený letoun bez regulace úhlu θ při konstantní nasta- vení nulové výchylky výškového kormidla	49
6.4	Výstup pro letoun s regulací úhlu θ	49
6.5	Zapojení regulátoru úhlu θ do systému FDC	50
A.1	59
A.2	60
A.3	61
B.1	62
B.2	63
B.3	63
B.4	64
B.5	64
B.6	65
B.7	66
B.8	67
B.9	67

SEZNAM TABULEK

1.1	Členy soustavy 1.9	13
1.2	Členy soustavy 1.10	14
5.1	Rozdělení souborů tvořících program VDVL do složek podle jejich funkce	41
5.2	Výpočet bezrozměrných derivací ve třídě <i>dimensionless_derivative</i> . .	44
5.3	Funkce pro výpočet bezrozměrných derivací ve třídě <i>dimensionless_derivative</i>	45
5.4	Výpočet rozměrových derivací ve třídě <i>dimensional_derivative</i>	45

SEZNAM PŘÍLOH

A Seznam zkratek v programu VDVL	59
B Tutorial k programu VDVL	62

A SEZNAM ZKRATEK V PROGRAMU VDWL

Parametry letounu			
Matlab	Hodnota	Popis	Jednotka
m	m	hmotnost	[kg]
ly	I_y	moment setrvačnosti k ose y	[kg*m ²]
cAK	c_{AK}	střední aerodynamická těživa	[m]
SK	S_K	plocha křídla	[m ²]
xTdsh	\bar{x}_T	poloha těžiště v %/100 st. aer. těživy	[-]
xAdsh	\bar{x}_A	poloha aerod. středu v %/100 st. aer. těživy	[-]
AK	A_K	zúžení křídla	[-]
CLK_alp	$C_{LK\alpha}$	sklon vztlakové čáry křídla	[rad ⁻¹]
CLVOP_alp	$C_{LVOP\alpha}$	sklon vztlakové čáry VOP	[rad ⁻¹]
SVOP	S_{VOP}	plocha VOP	[m ²]
kVOP	k_{VOP}	koeficient snížení kinetického tlaku v místě VOP	[-]
IVOP	I_{VOP}	rameno VOP k 0,25 stř. aerod. těživy	[m]
eps_alp	$\frac{\partial \varepsilon}{\partial \alpha}$	derivace zešikmení proudu v místě VOP podle úhlu náběhu	[-]
e0	e_0	Oswaldův koeficient	[-]
CD0	C_{D0}	součinitel odporu při nulovém úhlu náběhu KT	[-]
n	n	násobek	[-]
alp_dltV	$\frac{\partial \alpha}{\partial \delta_V}$	derivace úhlu náběhu podle změny výchylky výškového kormidla	[-]
Referenční hodnoty			
Matlab	Hodnota	Popis	Jednotka
VR	V_R	referenční rychlost	[m/s]
ThR	Θ_R	referenční úhle Θ	[°]
RhoR	ρ_R	referenční hustota vzduchu	[kg/m ³]
gmmR	γ_R	referenční úhel klesání	[°]

Obr. A.1:

Bezrozměrné derivace			
Matlab	Hodnota	Popis	Jednotka
CX_u	C_{x_u}	Bezrozměrová derivace dle rychlosti v podélném směru	[-]
CX_q	C_{x_q}	Bezrozměrová stabilitní derivace dle rychlosti klopení	[-]
CX_alp	C_{x_α}	Bezrozměrová derivace dle úhlu náběhu	[-]
CX_alpdot	$C_{x_{\dot{\alpha}}}$	Bezrozměrová derivace dle rychlosti změny úhlu náběhu	[-]
CX_dltV	$C_{x_{\delta_V}}$	Bezrozměrná řiditelnostní derivace dle výchylky výškového kormidla	[-]
CX_dltVdot	$C_{x_{\dot{\delta}_V}}$	Bezrozměrná řiditelnostní derivace dle rychlosti vychylování výškového kormidla	[-]
CZ_u	C_{z_u}	Bezrozměrová derivace dle rychlosti v podélném směru	[-]
CZ_q	C_{z_q}	Bezrozměrová stabilitní derivace dle rychlosti klopení	[-]
CZ_alp	C_{z_α}	Bezrozměrová derivace dle úhlu náběhu	[-]
CZ_alpdot	$C_{z_{\dot{\alpha}}}$	Bezrozměrová derivace dle rychlosti změny úhlu náběhu	[-]
CZ_dltV	$C_{z_{\delta_V}}$	Bezrozměrná řiditelnostní derivace dle výchylky výškového kormidla	[-]
CZ_dltVdot	$C_{z_{\dot{\delta}_V}}$	Bezrozměrná řiditelnostní derivace dle rychlosti vychylování výškového kormidla	[-]
Cm_u	C_{m_u}	Bezrozměrová derivace dle rychlosti v podélném směru	[-]
Cm_q	C_{m_q}	Bezrozměrová stabilitní derivace dle rychlosti klopení	[-]
Cm_alp	C_{m_α}	Bezrozměrová derivace dle úhlu náběhu	[-]
Cm_alpdot	$C_{m_{\dot{\alpha}}}$	Bezrozměrová derivace dle rychlosti změny úhlu náběhu	[-]
Cm_dltV	$C_{m_{\delta_V}}$	Bezrozměrná řiditelnostní derivace dle výchylky výškového kormidla	[-]
Cm_dltVdot	$C_{m_{\dot{\delta}_V}}$	Bezrozměrná řiditelnostní derivace dle rychlosti vychylování výškového kormidla	[-]

Obr. A.2:

Rozměrové derivace			
Matlab	Hodnota	Popis	Jednotka
X_u	X_u	Rozměrová stabilitní derivace X	$[N/(ms^{-1})]$
X_w	X_w	Rozměrová stabilitní derivace X	$[N/(ms^{-1})]$
X_q	X_q	Rozměrová stabilitní derivace X	$[N/(rads^{-1})]$
X_wdot	$X_{\dot{w}}$	Rozměrová stabilitní derivace X	$[N/(ms^{-2})]$
X_dltV	X_{δ_V}	Rozměrová stabilitní derivace X	$[N/(rad)]$
X_dltVdot	$X_{\dot{\delta}_V}$	Rozměrová stabilitní derivace X	$[N/(rads^{-1})]$
Z_u	Z_u	Rozměrová stabilitní derivace Z	$[N/(ms^{-1})]$
Z_w	Z_w	Rozměrová stabilitní derivace Z	$[N/(ms^{-1})]$
Z_q	Z_q	Rozměrová stabilitní derivace Z	$[N/(rads^{-1})]$
Z_wdot	$Z_{\dot{w}}$	Rozměrová stabilitní derivace Z	$[N/(ms^{-2})]$
Z_dltV	Z_{δ_V}	Rozměrová stabilitní derivace Z	$[N/(rad)]$
ZdltVdot	$Z_{\dot{\delta}_V}$	Rozměrová stabilitní derivace Z	$[N/(rads^{-1})]$
M_u	M_u	Rozměrová stabilitní derivace M	$[Nm/(ms^{-1})]$
M_w	M_w	Rozměrová stabilitní derivace M	$[Nm/(ms^{-1})]$
M_q	M_q	Rozměrová stabilitní derivace M	$[Nm/(rads^{-1})]$

Obr. A.3:

B TUTORIAL K PROGRAMU VDL

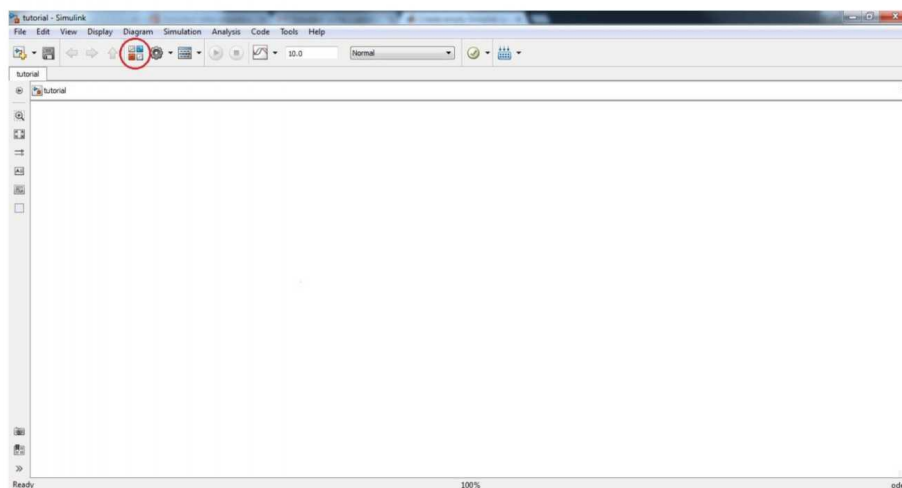
VDL

Tutoriál

1. Matlab Simulink

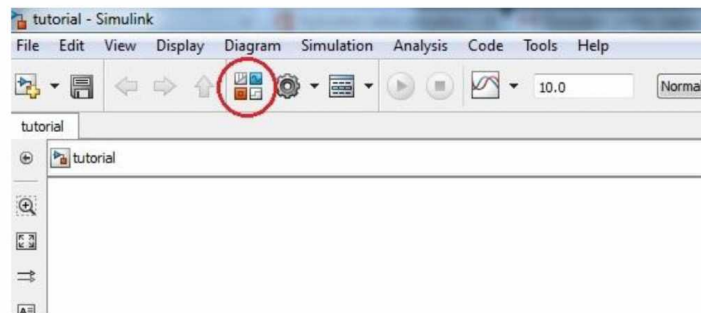
Pokud byl tento tutoriál otevřený přímo z programu VDL, pak se otevřel i čistý soubor Matlab simulink. Pokud se tak nestalo, v uživatelském rozhraní Matlabu napište příkaz: `new_system('název nového modelu')` a poté `open_system('název nového souboru')`, kde místo textu v uvozovkách v závorce napište jméno, pod kterým budete chtít nový model uložit.

Na obrázku 1 je prázdný dokument pro vytváření blokových diagramů, který se otevírá při spuštění spolu s tímto tutoriálem. V obrázku je vyznačená funkce „Library Browser“, kterou otevřete knihovnu bloků používaných při blokovém programování Simulinku.



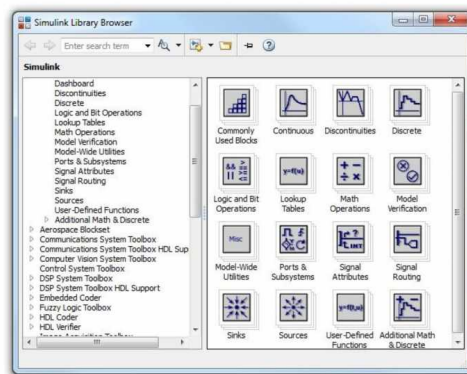
Obrázek 1 Prázdný Simulink dokument

Obr. B.1:



Obrázek 2 Detail - červeně označená funkce pro vyvolání Library Browser

Knihovna bloků obsahuje blokové funkce Simulinku, používané pro vytváření blokových modelů. Každý blok má svojí unikátní funkci a vlastnost. Bloky jsou rozřazeny do několika skupin, podle jejich vlastností. V případě, že máte nainstalované doplňkové knihovny pro Simulink jako je Aerospace Blockset, nebo různé toolboxy, zobrazují se v levém panelu. Pro účely tohoto tutoriálu budou využity pouze bloky ze základní knihovny Simulinku.



Obrázek 3 Knihovna blokových funkcí Simulinku

Jednotlivé části knihovny lze rozkliknout v postranním panelu, nebo ikonou v hlavní nabídce. V tomto tutoriálu budou použity následující bloky:

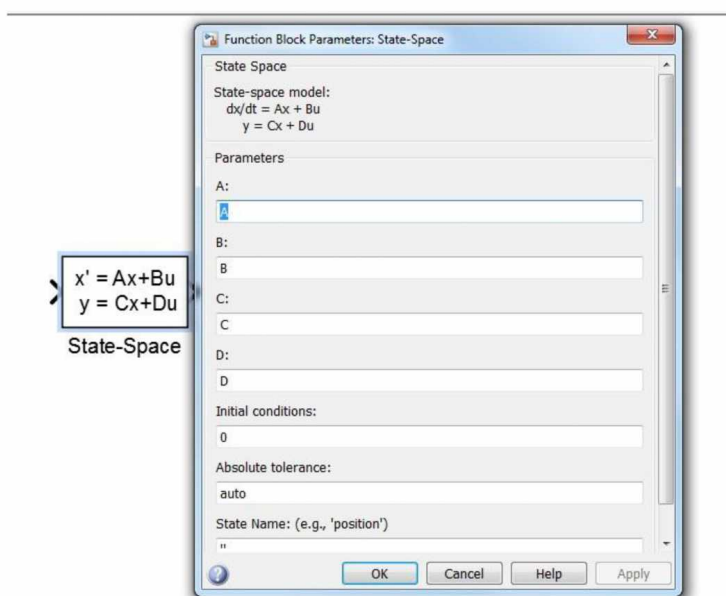
- Step z nabídky Sources
- Constant z nabídky Sources
- Add z nabídky Math Operations
- Divide z nabídky Math Operations
- State-Space z nabídky Continuous
- To workspace z nabídky Sinks
- Demux z nabídky Signal Routing

Obr. B.2:

2. Vytvoření základního modelu pro simulaci odezvy na výchylku výškového kormidla

Jednotlivé bloky se do pracovního prostoru jednoduše přetahují myší. V prvním kroku je třeba do prostoru přetáhnout blok *State-Space*. Dvojklikem se otevřou vlastnosti bloku. Blok State-Space umožňuje nastavení matic A, B, C a D stavového prostoru, počáteční podmínky a absolutní tolerance. Matice lze zadávat přímo v hranatých závorkách, nebo jako proměnné. Jsou-li v maticích nastavené názvy proměnných, pak tyto hodnoty Simulink vyhledává ve workspace Matlabu.

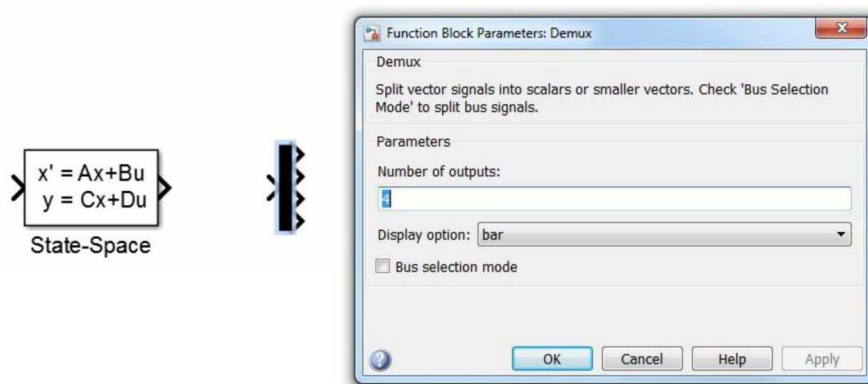
Vlastnosti bloku State-Space nastavte podle obrázku 4.



Obrázek 4 Nastavení bloku State Space

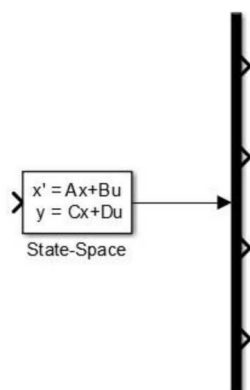
V dalším kroku budou nastaveny výstupy z bloku State-Space. Protože je podélný pohyb vyjádřen dvěma hodnotami rychlosti (u a w [m/s]), jednou hodnotou úhlového zrychlení (q [rad/s]) a jedním úhlem (θ [rad]), je třeba výstup rozdělit do čtyř vláken pomocí bloku *Demux*. Přetáhneme tedy tento blok do pracovního prostoru a v jeho vlastnostech nastavíme 4 výstupy.

Obr. B.3:



Obrázek 5 Vložení a nastavení bloku Demux do modelu

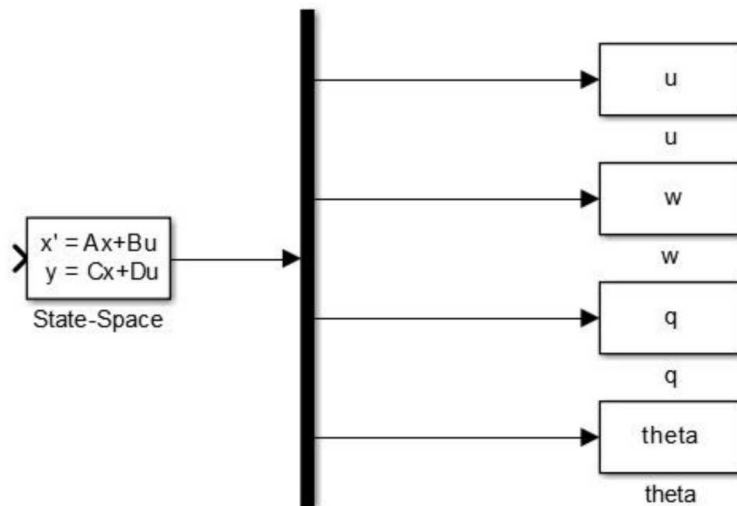
Velikost bloku lze měnit vytažením z jeho rohu. Velikost bloku *Demux* bude potřeba 3 – 4x zvětšit. Dále vytažením šipky z výstupu z bloku *State-Space* propojte tento blok s blokem *Demux*.



Obrázek 6 Propojení bloků

Vložte do pracovního prostoru čtyři bloky *To workspace* a propojte je s výstupy bloku *Demux* podle obrázku 7. V jejich vlastnostech nastavte název podle obrázku a výstupní formát *Timeserie*.

Obr. B.4:



Obrázek 7 Přidání a pojmenování bloků To Workspace

Nyní je třeba vytvořit vstupní impuls do stavového prostoru, reprezentovaného blokem *State-Space*. Bude vytvořena vstupní impulsní výchylka výškového kormidla, pomocí dvou bloků *Step* a jednoho bloku *Add*.

Blok *Step* vysílá po nastavenou dobu konstantní signál a ve zvoleném okamžiku skokově změní jeho hodnotu na požadovanou úroveň. Dále pokračuje na této hladině až do konce simulace. Pomocí dvou bloků lze nastavit konečné hodnoty bloků *step* a jejich časy tak, aby signál po sečtení výstupů z těchto bloků měl obdélníkový tvar. K sečtení signálů slouží blok *Add*.

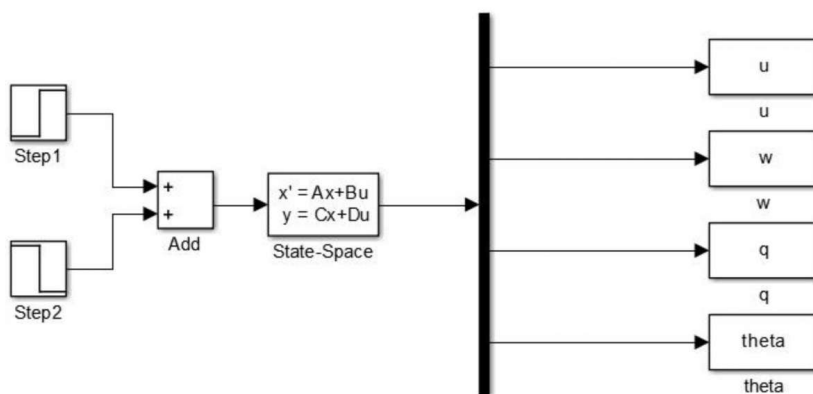
Impulsní výchylka kormidla se obvykle nastavuje tak, aby obsah vzniklého obdélníku byl roven jedné. Za předpokladu, že počáteční hodnota v čase je u obou bloků 0, blok 1 má nastavenou konečnou hodnotu V_k v čase t_1 a blok 2 v čase t_2 , pak je obsah obdélníků dán vztahem:

$$A = (t_2 - t_1) * (V_k)$$

Vytvořte vstupní výchylku podle obrázku 8 a nastavte *Step* bloky tak, aby byl obsah obdélníku roven jedné.

Obr. B.5:

Pozn.: Zadávané hodnoty do bloků step jsou v radiánech.



Obrázek 8 Vstupní signál

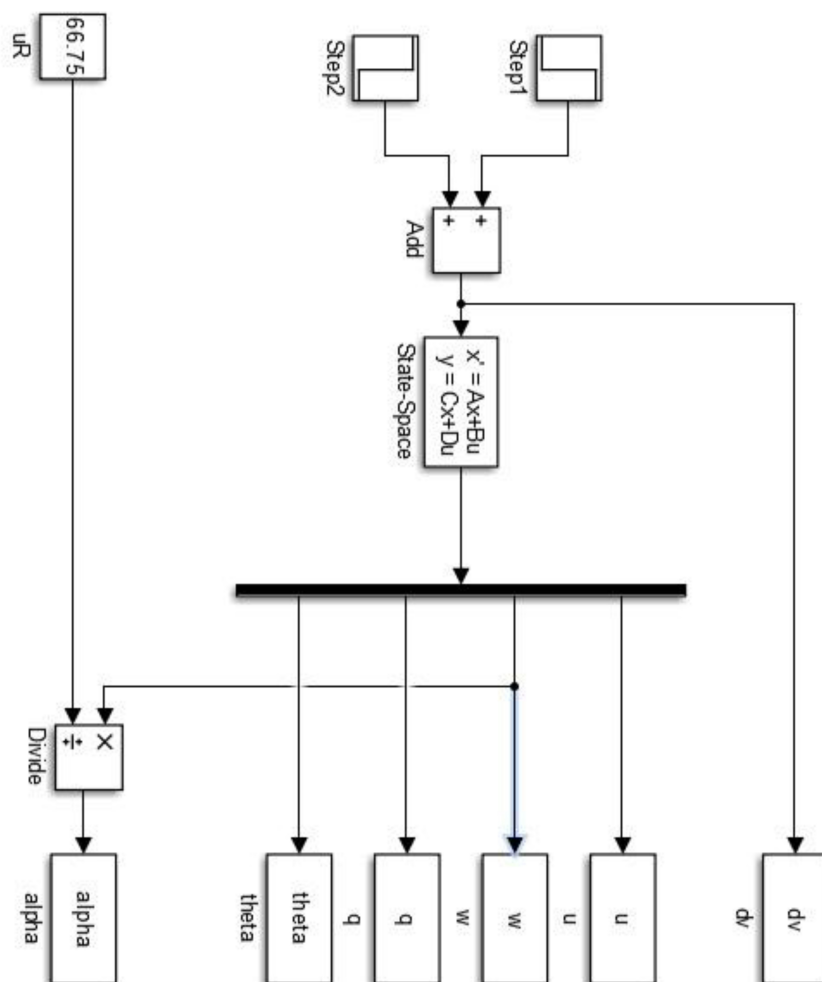
V dalším kroku přidejte do modelu bloky *Constant* a *Divide*. Blok *Constant* vytváří konstantní signál a bude použit pro získání referenční rychlosti letounu. Blok *Divide* provádí operaci násobení a dělení signálů. Pomocí těchto bloků lze vypočítat úhel náběhu letounu dle vzorce:

$$\alpha(t) = \frac{w(t)}{u_R}$$

Propojte bloky podle obrázku 9 a přidejte ještě jeden výstupní blok propojený s vlákem mezi blok *Add* a *State-Space* (blok bude získávat a ukládat signál výchylky kormidla do stavového prostoru).

Je důležité, aby názvy bloků *To workspace* a *Constant* byly pojmenovány správně, pokud budou výstupy posílány do výstupního okna programu VDVL.

Obr. B.6:



Obrázek 9 Konečná podoba model

Obr. B.7:

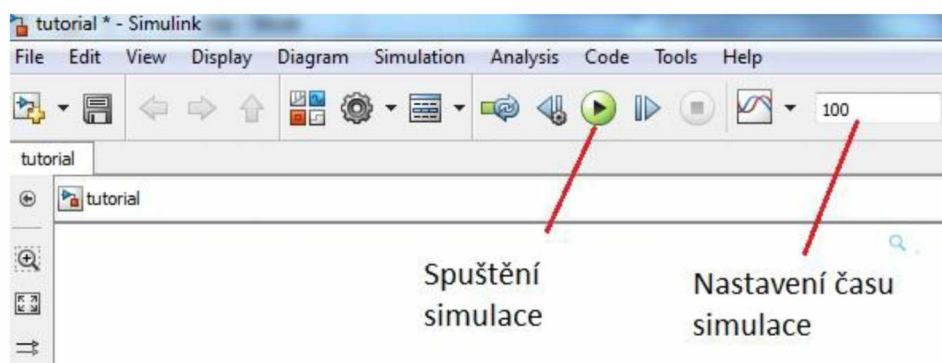
3. Spuštění simulace

Je-li vytvořený model podle předchozího návodu, je před jeho spuštěním nutné definovat matice A, B, C a D ve workspace, aby je mohl načíst blok *State-Space*. Program VDVL tyto matice vytváří tlačítkem „Nastavit model“, pokud jsou definovány v tabulce „Koeficienty matic A a B“. Koeficienty matic lze spočítat v Zadávacím okně, nebo je možné je zde načíst z databáze v poli „Akce“ (viz návod).

Pokud byly načteny hodnoty některého z letounů v databázi, je třeba uložit nastavení tlačítkem „Zadat“ v zadávacím okně. Po spuštění tlačítka „Nastavit model“ v základním okně programu pak budou vytvořeny příslušné matice ve *workspace* Matlabu.

Po nastavení matic je třeba v bloku *Constant* nastavit referenční rychlost letounu, pro kterou byly vypočítány matice stavového prostoru. Tu lze najít v tabulce „Referenční hodnoty“.

Pokud jsou všechny hodnoty řádně nastavené, lze spustit simulaci v horním panelu pracovního prostředí Simulinku. Dobu simulace je dobré nastavit přibližně na 50 sekund pro skokovou změnu (projevují se pomalé kmity) a na 10 sekund pro jednotkový impulz (projevují se rychlé kmity).



Obrázek 10 Nastavení simulace

Obr. B.8:

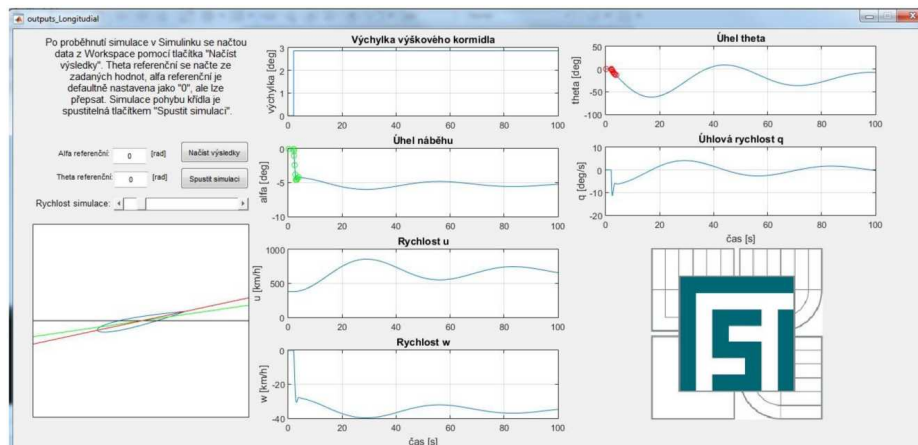
4. Výsledky simulace

Veškerá výstupní data ze simulinku byly přeneseny do workspace pomocí bloků *workspace* ve formátu Timeseries. Tento formát obsahuje dva sloupce, kde první tvoří čas od spuštění simulace a druhý jsou hodnoty dané veličiny v tomto časovém okamžiku.

Program VDVL dokáže výstupní hodnoty zobrazit ve výsledkovém okně přímo z workspace Matlabu. Po dokončení simulace otevřete výsledkové okno a spusťte tlačítko „Načíst výsledky“. Do grafů se vykreslí výstupní hodnoty v závislosti na čase. Hodnoty výstupu jsou odchylky od referenčních hodnot. Referenční úhly α a θ lze zadat přímo ve výsledkovém okně a po načtení výsledků se k odchylkovým hodnotám přičtou. Pro zjištění skutečné rychlosti v závislosti na čase by bylo třeba upravit model v Simulinku přidáním dalšího bloku *Add* a k výstupní hodnotě odchylky podélné rychlosti přičítat referenční rychlost z bloku *Constant*.

Po načtení výsledků do výstupového okna je možné spustit simulaci pohybu křídla vůči vzduchu a horizontu pomocí tlačítka „spustit simulaci“. Černá čára v simulaci reprezentuje horizont, zelená pak směr nabíhajícího proudu na křídlo a červená čára je prodloužená tětíva profilu. Rychlost simulace lze regulovat přetažením posuvníku oknem s probíhající simulací.

V simulaci se profil natáčí podle úhlu theta, zelená čára rotuje kolem profilu podle úhlu alfa a samotný profil se vertikálně pohybuje, podle změny výšky během simulace.



Obrázek 11 Výsledkové okno se spuštěnou grafickou simulací

Obr. B.9: